WARNING:
Amiga Virus!
We've Got The Fix

Amazing COMPUTING™

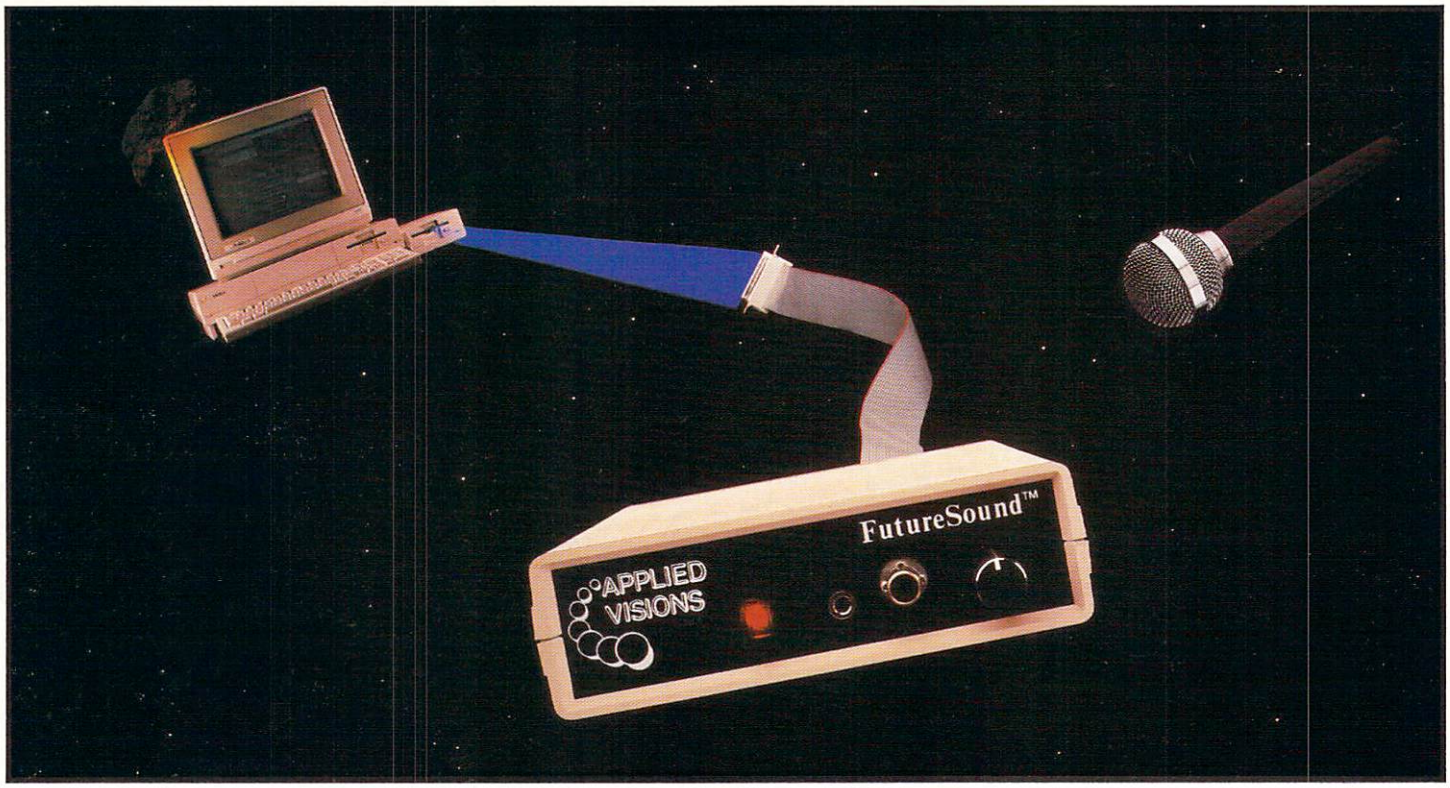
Your Original AMIGA™ Monthly Resource

Volume 2 Number 12
US $3.50 Canada $4.50

AMIGA Desktop Video

An Amazing Mini-Series

Inside WordPerfect



"Open the pod bay doors, HAL..."

Programmers cast their vote!

Right now, leading software developers are hard at work on the next generation of Amiga® products. To add the spectacular sound effects we've all come to expect from Amiga software, they are overwhelmingly choosing one sound recording package... FutureSound. As one developer put it, "FutureSound should be standard equipment for the Amiga."

FutureSound the clear winner...

Why has FutureSound become the clear choice for digital sound sampling on the Amiga? The reason is obvious: a hardware design that has left nothing out. FutureSound includes two input sources, each with its own amplifier, one for a microphone and one for direct recording; input volume control; high speed 8-bit parallel interface, complete with an additional printer port; extra filters that take care of everything from background hiss to interference from the monitor; and of course, a microphone so that you can begin recording immediately.

What about software?

FutureSound transforms your Amiga into a powerful, multi-track recording studio. Of course, this innovative software package provides you with all the basic recording features you expect. But with FutureSound, this is just the beginning. A forty-page manual will guide you through such features as variable sampling rates, visual editing, mixing, special effects generation, and more. A major software publisher is soon to release a simulation with an engine roar that will rattle your teeth. This incredible reverberation effect was designed with FutureSound's software.

APPLIED VISIONS

Question: What can a 300 pound space creature do with these sounds?

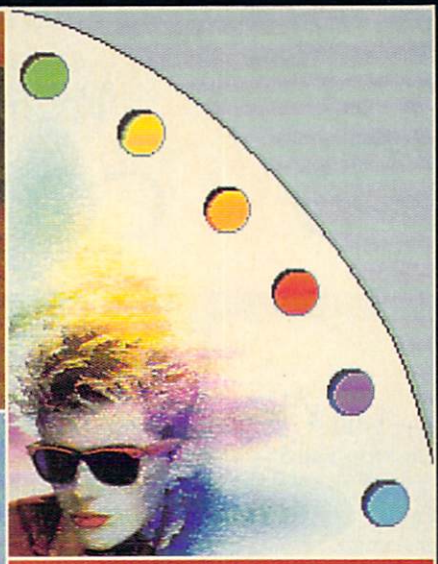
Answer: Anything he wants.

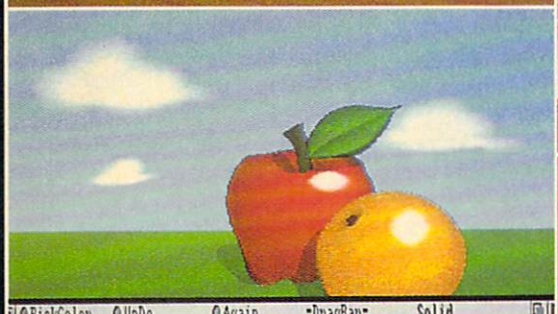
Since FutureSound is IFF compatible (actually three separate formats are supported) your sounds can be used by most Amiga sound applications. With FutureSound and Deluxe Video Construction Set from Electronic Arts, your video creations can use the voice of Mr. Spock, your mother-in-law, or a disturbed super computer.

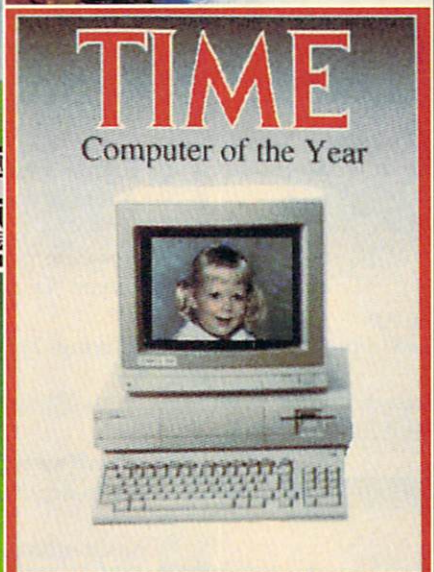
Programming support is also provided. Whether you're a "C" programming wiz or a Sunday afternoon BASIC hacker, all the routines you need are on the non-copy protected diskette.

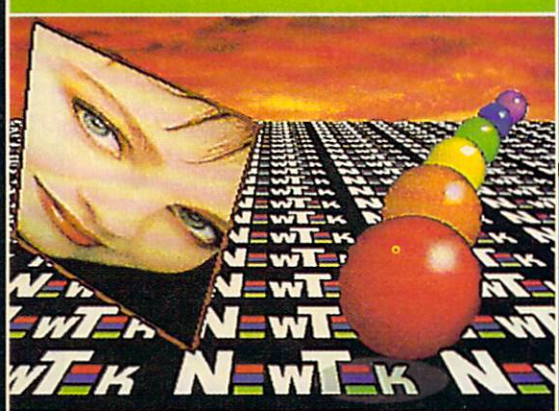
Your Amiga dealer should have FutureSound in stock. If not, just give us a call and for $175 (VISA, MasterCard or COD) we'll send one right out to you. Ahead warp factor one!

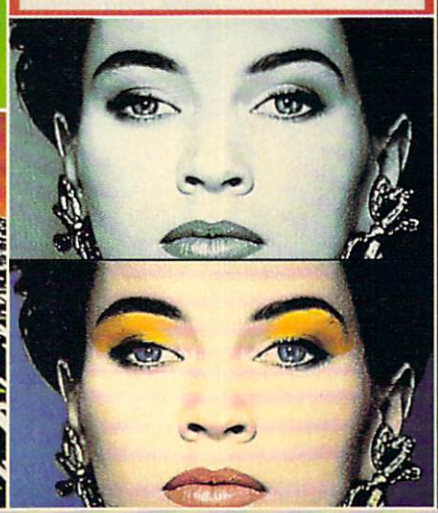
Applied Visions, Inc., Suite 2200, One Kendall Square
Cambridge, MA 02139 (617) 494-5417











ONLY DIGI PAINT CAN DO ALL THIS

Get the maximum graphics power from your Amiga. Create stunning, lifelike computer artwork with Digi-Paint, the first full-featured 4096 color (Hold and Modify) paint program. Break the "32 color barrier" and finally realize the potential of your Amiga with Digi-Paint's advanced features:

- 4096 colors on screen simultaneously
- NewTek's exclusive enhanced HAM mode
- Dithered HAM gradient fill
- Full screen effects including double, half size, mirror reverse and more
- Full IFF and Digi-View compatibility
- Use 320x200 or HAM hi-res 320x400 resolutions
- Fat bits Magnify mode
- Rectangle, oval, line and other drawing tools
- 12 different paint modes including blending, tinting and smooth shading
- Full lasso cut and paste with automatic edge blending
- Programmed completely in assembly language for fast, smooth response

Find out why Byte Magazine called Digi-Paint "Remarkable". Available now at your local Amiga dealer or call: **1-800-843-8934.**

ONLY $59.95

NEWTEK INCORPORATED

Amazing COMPUTING™

Volume 2, Number 12

CONTENTS

Amazing Features

Amazing Reviews

Amazing Columns

Amazing Departments

Amazing COMPUTING™

Publisher:	Joyce Hicks
Circulation Manager:	Doris Gamble
Asst. to the Publisher:	Robert James Hicks
Corporate Trainer:	Virginia Terry Hicks
Traffic Manager:	Robert Gamble
Managing Editor:	Don Hicks
Submissions Editor:	Ernest P. Viveiros Jr.
Hardware Editor:	Ernest P. Viveiros Sr.
Music & Sound Editor:	Richard Rae
Art Director:	Keith M. Conforti
Advertising Manager:	John D. Fastino
Senior Copy Editor:	Michael T. Cabral
Copy Editor:	Karen P. Granger
Amicus & PDS Editor:	John Foust
Production Manager:	Mark Thibault
Production Assistant:	Rico A. Conforti
Production Assistant:	Keven Desmarais

Advertising Sales & Editorial
1-617-678-4200

Special thanks to:
Lynn Hathaway
Donna Peladeau
Traci Desmarais
Pilar Medeiros
Donna Thibault
Betsy Piper at Tech Plus.

Amazing Computing™ (ISSN 0886-9480) is published by PiM Publications, Inc., P.O. Box 869, Fall River, MA 02722. Subscriptions: In the U.S. 12 issues for $24.00; in Canada & Mexico, $30.00; Overseas; $35.00. Printed in the U.S.A.

First Class or Air Mail rates available upon request.

PiM Publications, Inc. maintains the right to refuse any advertising.

PiM Publications, Inc. is not obligated to return unsolicited materials. All materials requesting return must be received with a Self Addressed Stamped Mailer.

From The Editor:

Great News!
It appears the large projection of Amiga™ sales by the Commodore Business Machine marketing staff may come true. Early reports show a great demand for Amiga 500™ and Amiga 2000™. For several weeks, Commodore has been out of stock on Amiga 2000™ and is hurriedly producing more. This shortage occured in October, months before the official Christmas panic should have started.

Amiga 500s, apparently in good supply, are also producing record sales. I visited a small store with limited traffic potential, only to see a stack of over 30 Amiga 500's. When I asked the clerk if they were stocking up for a projected shortage, he said, "No, that's just a weeks supply."

If you are waiting for an order from your dealer, be kind and patient. Commodore assures us they are doing all they can to produce the material we need.

Good News!
AmiExpo™ in New York City was well attended and appreciated by new and old Amiga enthusiasts alike. It was a particular pleasure to be able to meet subscribers, dealers and Amiga developers face-to-face for the first time. The seminars were well-attended and the exhibitors enjoyed wall-to-wall attendance at their booth demonstrations. For a deeper look at the show, please see John Foust's "Amicus" column.

Amiga Enthusiasts' Generosity
As is often the case at such shows, we wanted to distribute magazines. As is the custom, the magazines should be given away free. However, we were concerned with the cost of shipping and handling, as well as the ability to control the number of issues distributed.

The eventual decision was to give a magazine away free for a contribution to the American Cancer Society. We expected nickles, dimes, and quarters. All the attendees were generous, with some giving five and ten dollars. The end result was a collection of $1215.77. PiM Publications Inc. is extremely grateful to all who gave. Let us hope this style of marketing/charity will spread. A great deal of good could come from this approach.

Some Bad News- A Second Amiga Virus Has Been Reported!
A Eurpean organization called SCA claims responsibility for incorporating a new virus into the Amiga community. This software virus is transmitted from Amiga to Amiga by infected disks.

Although this prank was designed to be humorous, the results have been disastrous for Amiga 2000™ software. One developer reportedly lost eight months of research and development when the virus found its way into the source code and all backups of a very costly project.

SCA claims responsibility--but is this the correct phrase? Their actions have created a situation where every Amiga disk is a potential "time bomb." These actions, however innocently conceived, have an effect similar to saying "I didn't know the gun was loaded." The Amiga community is more mature and of better nature than to accept a blatant attack on their systems.

In this issue, we have included a small article and a "fix" to be applied before the virus has had time to seriously infect your software. However, this is a fix and not a cure. As we go to press, there is a rumor of a public domain software program to inspect suspected disks for the virus. As soon as such a program is available, it will be included in the Amicus Disk collection.

Best Wishes
Last, but first in the minds of the staff and management at *Amazing Computing*™, are our Best Wishes to our readers, dealers, writers, developers and their families for a very Happy Holiday.

Best Wishes

Don Hicks
Managing Editor

Amazing Dealers

The following are Amazing Dealers, dedicated to supporting the Commodore-Amiga™. They carry Amazing Computing™, your resource for information on the Amiga™.

If you are not an Amazing Dealer, but would like to become one, contact:

PiM Publications, Inc.
P.O.Box 869
Fall River, MA. 02722
1-617-678-4200

Alabama
Abax Data Systems — Huntsville
Alabama Book Store — Tuscaloosa
Command Computers — Birmingham
Electronics Boutique — Hoover
Mail Tel Communications — Mobile
Mel's Photo & Computer Shop — Montgomery
Universal Computer System — Mobile

Arkansas
Computermart — Mountain Home
The Micro Shop — Little Rock
The 64 Store — Drytheville
The 64 Store — Jonesboro
The 64 Store — North Little Rock

Arizona
Computer Library, Inc. — Tempe
Computer West — Phoenix
Computer Works — Glendale
Copperstate Business Systems — Phoenix
Hacko's — Tempe
North American Digital — Tucson

California
Aday International — Foster City
Augnet — Fremont
Balboa Office Supply — San Diego
Bits n Pieces Comp Prod. — Anaheim
Brown Knows Computing — Redlands
Byte Factory Computers — Salinas
Century Computer Systems — La Habra
Circus of Books — Los Angeles
Circus of Books — Los Angeles
Commodore Land — Alhambra
Commosch — National City
Compunik — Los Angeles
Computer Attic — Palo Alto
Computer Cellar — Anaheim
Computer Nook, Inc. — San Bernadino
Computer Outlet — San Diego
Computer Outlet — San Marcus
Computer Time — Citrus Hts
ComputerLand Fremont — Fremont
Coronet Newsstand — Oceanside
Dataphile — Watsonville
Delta Spectrum — Laguna Beach
Discount Computer Station Inc. — Long Beach
Empire Computers — Santa Rosa
ECX Computer Co. — Walnut Creek
Harding Way News — Stockton
Heinrichson, Schneider & Young, Inc. — Berkeley
Home Computing Center — Concord
Home Computing Center — Mill Valley
Home Computing Center — San Leandro
Home Computing Center — San Bruno
HT Electronics — Sunnyvale
K J Computers — Granada Hills
Lavity — North Hollywood
LTC Computers — Santa Ana
Micro Flight — San Jacinto
MTS Inc. Tower Books — Sacramento
Nonagon Software — El Cerrito
Oo Amp Tech Books — Los Angeles
Paras Newsstand — San Diego
Pat's Electronics — Rio Linda
Real Computers & Electronics — Petaluma
Ridgecrest Computer Ctr. Inc. — Ridgecrest
Sawyer's News Inc. — Santa Rosa
S.O.S. Computers — Los Angeles
Software First — Santa Rosa
Software Cellar — Anaheim
Software Central — Pasadena
Software Plus — Citrus Heights
The Booksmith — San Francisco
Tower Books — Chico
Tower Books — Concord
Tower Books — Mountain View
Tower Books — Sacramento
Tower Books — Sacramento
Tower Books — Sacramento
Tower Books — San Mateo
The Computer Room — Scotts Valley
Warner Engineering — San Diego
Winner's Circle — Berkley
World Book & News — Hollywood

Colorado
Aurora Newsstand — Aurora
Book Tech Distributing — Englewood
Citadel Computers — Colorado Springs
Computer Discount — Denver
Computer Room — Aurora
Ebert Personal Computers — Aurora
Micro Computer World — Fort Collins
Micro World — Lakewood
Sun Country Computers — Colorado Springs

Connecticut
Aetna Life Club Store — Hartford
Amigo Business Computer — Norwalk
Computer Outlet — PlaVernon
Friendly Computers — Plantsville
Mnemonics Inc. — Norwalk
Personal Computer Center — Norwich
Software City — Orange
Software Kingdom — East Windsor
Spectrum Computers — New Haven
Tricom Computers — North Windham

Delaware
Castle Video — Newark

Florida
A A Computers — Jacksonville
Commodore Country Inc. — Pinellas Park
Computer Bar — Pennsacola
Computer Base — Brandon
Computer Image — Miami
Computer Specialty — Melbourne
Computer Terminal Inc. — Sebring
Computers 4 Rent — W. Palm Beach
Computers Plus — S. Daytona
Cotronics Inc. — Stewart
Data Base — Davie
Education Computers Etc. — Talahasse
Family Computers — Sarasota
Florida Book Store — Gainesville
Heartland Computers — Mulberry
Joe's News, Inc. — Miami
K&L Computers — E. Palatka
Megaport Computer Center — Venice
Micro Computer Center — Largo
New Age Electronics — St. Petersburg
Program Store — Clearwater
Random Access Computers Inc. — Ft. Walton Beach
Software City — Orange Park
Software Software — Plantation
Sunny's At Sunrise, Inc. — Sunrise
The Open Door — Cocoa
The PC Collection — Coral Gables

Georgia
Future System — Augusta
Software South Inc. — Savannah
The 64 Store — Atlanta

Hawaii
Astro Computers & Electronics — Aiea
Computer House Inc. — Honolulu
J & C Repair — Aea

Idaho
A.B.I. Computer & Video — Nampa
Boise Computer & Video Inc. — Boise

Illinois
A Bitchester — Rockford
Computerland O'Hare — Park Ridge
ComputerLand of Niles — Niles
DigitalWorld — Addison
I.C.V. Data Center Inc. — Mundelein
Illini Microcomputers Inc. — Naperville
Krech's & Brentano's — Chicago
Micro Age — Oaklawn
Software Or Systems — Fairview Hts.
Software Plus — Wheeling
Software Plus West — Hanover Park
The Memory Expansion — Bloomington
Unique Computer — Sterling

Indiana
Burkat Computer Ctr. — South Bend
Bytrex of Angola — Angola
Computer Corner — Fort Wayne
Computer Corner — Marion
Computer People Inc. — Michigan City
Greg Chaney — Greenfield
Indiana News — Indianapolis
Micro Computer Centers Inc. — Indianapolis
Spectra Computer Sys. Inc. — Evansville
Vons Computers — West Lafayette

Iowa
Gronet Computers — Des Moine
Micro Computer Applications LTD — Marshalltown

Kansas
J & L Electronics — Liberal
MidKansas Computers — Newton
Midwest Computers — Manhattan
Software City — Overland Park

Kentucky
Computer Dimensions Inc. — Lexington
Computer Store of Kentucky — Louisville
Hawley Cooke — Louisville
Reality World Computers — Lexington

Louisiana
Computer Time — Shreveport
Modern Business Machines — Metairie
Software Center International — Metairie
Software Solutions — Baton Rouge
Software Source — Lafayette
The Box Shelf — Slidell
The Computer Clinic — Lafayette

Maryland
Buried Treasure — Rockville
Computerware TODAY — Gaithersburg
Compuvision Vision Center — Columbia
Layhill Newsstand — Kensington
Micro Computer Center — Baltimore
Software Advantage — Rockville
Software N Things — College Park
Waldorf Computer — Waldorf

Massachusetts
Club Computer — Cambridge
D-Five — Bedford
DOX — Williamstown
General Computer Store — Framingham
HCS Computer Center — Marshfield
LCA Video & Computer Ctr. — Norwood
Memory Location — Wellsley
OmniTek Computers — Tewksbury
Pioneer Valley Data Equipment — Amherst
Softpro — Burlington
The Bit Bucket — West Newton
The Software Haus — Quincy
Tycom Inc. — Pittsfield

Maine
Computer Barn — Damariscotta
Hands On Computing — Westbrook
Radio Shack Associate Stores — Jay

Michigan
Basic Computer Center — Grand Rapids
Bits-Bytes-Nibbles — Petowskey
Canton Computer — Canton
CompuSoft — Kalamazoo
Direct Access — Lansing
Edmunton Computer Center — Pontiac
Global Computer Center — Dearborn
Central Park Computers — Okemos
Michigan Software — Novi
Midland Computer Shop — Midland
Pro-Video — Okemos
Programs Unlimited — Grand Rapids
Readmore Newscenter — Lansing
Roseville Computer Store — Warren
Slipped Disk — Madison Heights
Specialists Computer — Livonia
State Street Computers — Ann Arbor
T.J. Enterprises — Clinton
The Edge Connector — Clio
The Soft Warehouse — Kalamazoo
Ye Olde Computer Shoppe — Ypsilanti

Minnesota
Computers, etc. — Eagan
JMH Software — Maple Grove
MCD of Hibbing Inc. — Hibbing
Midwest Computer Services — Clearwater
Shinder's Read More — Minneapolis
Specialists IN — Hopkins
Valiant Inc. — Stillwater

Mississippi
National Computer Center — Ocean Springs

Missouri
Associated Computer Services — Springfield
Brands Mart Computers — Kansas City
Data Byte Computer Co. — St. Louis
Instant Replay Ltd. — Chesterfield
Systems Plus — Independence
Systems Plus of St. Louis — St. Louis

Montana
Appelgren Computer Systems — Great Falls

Nebraska
BULLS-EYE — Lincoln
My Com Inc. — Plattsmouth
Double E Electronics — Omaha

Nevada
Century 23 Inc. — Las Vegas
Computer World — Las Vegas
Tower Records — Las Vegas

New Hampshire
Diversified Computers — Keene

New Jersey
Family Computer Centres — South Orange
Morris County Stationers — Chester

New Mexico
New Horizon Computer System — Alamogordo
Page One Newsstand — Albuquerque
Technological Concepts — Las Cruces

New York
Amuse — New York
Amicus- Hudson Valley — Wappingers Falls
Amiga Business Computer — North Port
Byte Shop — Merrick
Compurite Center — White Plains
Computer Outlet — Jamestown
Computer Palace — Felden
Computers Etc. — Syracuse
Consumers Software — Commack
Crainland Inc. — Brooklyn
Leights Computers — New York
Mnemonics — North Port
N.U.B.S. Computer Center — Bronx
Software Centre International — Forest Hills
Software & Such — Scotia
Software City — Forest Hills
Software Supermarket — Tonawanda
Star Tech Systems — Massena
Synfaire Trading — New York
Tadiran — New York
Tedrow Business Products — Rochester
The Computer Cellar — Albany
The Working Computer — Stoney Brook
Video Computer Bin — Brooklyn
Video Computer Center — Rome
Video Connection — Schenectady
Village Computer Inc. — New York
World Computers Inc. — Hicksville

North Carolina
C2 of Hickory — Hickory
Digitz — Raleigh
K&S Newsstand — Clemmons
K&S Newsstand — Kernersville
K&S Newsstand — Winston-Salem
K&S Newsstand — Winston-Salem
K&S Newsstand — Winston-Salem
Software City — Raleigh
Software City — Charlotte
TDS Computers — Durham
Triad Computers — Winston-Salem

North Dakota
Computer Associates — Fargo
G. F. Computers — Grand Forks
The Computer Store — Minot

Ohio
Computer Network — Newark
Computer Site — Cleveland
Earthrise Micro Systems — Columbus
Fairborn Home Computer — Dayton
Lakes Consumer Electronics — Akron
Main Office Supply — Coshocton
Micro Center — Columbus
Microwave Magic — Fairfield
North Coast Programming — Wiloughby
Personal Computer Co. — Youngstown
Paper & Ink — Tiffin
Quality Computer Application — Toledo
Saxtan Computer Center — Sandusky
Software Centre International — North Olmsted
Software and More — Cincinnati
Software City — Centerville
Software Connection — Mayfield Hts.

Oklahoma
Colonial Video & Computer — Bartlesville
Micro Link Computer Store — Tulsa
P C Tech — Tulsa
Second Hand Software — Oklahoma City
Video-Comp Inc. — Lawton

Oregon
Andromeda — LaGrande
Capitol News Center — Salem
Clackamas Computers — Beaverton
Clackamas Computers — Clackamas
Computron Business Systems — Portland
Powell's Books — Portland
Powell's Tech — Portland
Rich Cigar Store — Portland
Software Express — Eugene

Pennsylvania
Alpha Omega EBE — Warren
Basic Computer Systems — Hermitage
Chester County Book Co. — West Chester
Circle Computer — Ephrata
Downington Computer Center — Downington
East Coast Software — Hershey
Floppy Disk — Mechanicsburg
Gene's Books — King of Prussia
North Central Digital Systems — Danville
Pittsburgh Computer Store — Pittsburgh
Triangle Computers — Indiana

Rhode Island
Micro Limits — Smithfield
Micro Limits — Warwick
Software Connection — Warwick

South Carolina
Avcom International Inc. — West Columbia
Horizon — Greenville
Micro Computer Depot — Sumter
Software City — Spartanburg
Watson's Jewlery — Chesterfield

South Dakota
Computer Business Center — Rapid City

Tennessee
ComputerLab of Memphis — Memphis
The 64 Store — Bartlett

Texas
Bookstop — Houston
Byte Shop — Austin
C.E.M. Corporation — Brazoria
Colonel Video — Houston
Computer Age — Houston
Computer Age — Dallas
Computer Magic — Austin
FNS Computer Services — Texarkana
Metropolitan Computer — Richardson
Micro Search — Houston
Software Library — Witchita Falls
Software Terminal — Fort Worth
The Computer Experience — San Antonio
The Computer Stop — Abilene

Utah
Computers Plus — Murray
Magazines and Gifts — Salt Lake City
People's Computers — Provo

Vermont
Ray Supply Inc. — South Burlington

Virginia
Alonso — Alexandria
Diskovery — Falls Church
Family Computer Center — Fairfax
I/O Computer — Norfolk
Software Central — Norfolk
Software City — Richmond
Virginia Micro Systems — Woodbridge

Washington
A.P.P.L.E. CO-OP — Renton
Apple Valley Computers — E. Wenatchee
Bellingham Computer Center — Bellingham
Bits, Bytes and Nibbles — Spokane
Bull Dog News — Seattle
Byte World — Kent
Computer Exchange Services — Bellingham
Computer Quest — Edmonds
Computers + Inc. — Tacoma
Computers and Movie Magic — Yakima
Family Computers — Lynwood
KAOS — Northeast Poulsbo
Nybbles & Bytes — Tacoma
Omni International — Seattle
Programs Plus — Seattle
Real Time Systems — Seattle
Take A Byte — Kennewick

Wisconsin
Colortron Computers — Racine
Computer Software Center — Milwaukee
Fox Valley Personal Computer — Oshkosh
Mom's Computers — Stevens Point
Richard Reske — Madison
TMW Software Inc. — Wausau

Wyoming
Byte Emporium — Laramie

Canada
Compute or Play — Alberta
Computerworks — Alberta
Computer Shop — Alberta
Data Pro — Alberta
Computer Shop of Calgary — Alberta
The Software House — Alberta
Hindsin Computing — Alberta
Red Deer Micro Systems — Alberta
Kamloops Land — British Columbia
Computer King — British Columbia
Genl — British Columbia
Nutec — British Columbia
Sprita — British Columbia
Strider — British Columbia
Super Software — British Columbia
Canadian Computer Era — Manitoba
Micromart — Manitoba
ComputerWorld — Manitoba
J.W. Allen — Manitoba
Kobetak — Manitoba
Minicomp — Manitoba
Arkon — Ontario
Chatam Computer — Ontario
Computerland — Ontario
Comspec — Ontario
Electronics 2001 — Ontario
Hastings Data — Ontario
The Computer & You — Ontario
Thornhill — Ontario
Software Co. — Quebec
Daltronic Solare — Quebec
Mierodata — Quebec
Info Mercier — Quebec
Laval Micro — Quebec
Vision Tronique — Quebec
Distribution Visipro — Quebec

Distributed in Canada by
Ingram Software Distributors — Concord, Ontario
Compulit Distributors — British Columbia
Phase 4 Dist. Inc. — Calgary, Alberta

Argentina
Logic Computer — Buenos Aires

Australia
Computermart — Leaderville,WA
Diskworks — Greenwich, NSW
High Technology Computer Supplies — Brighton, Melborn
John Verhoven — Cloverdale, WA
Maxwell Office Equipment — Abbotsford, Melburn
Megadisk — Lane Cove
Megatronics — Lanecove,NSW

Saudia Arabia
Abdulla Fouad & Sons — Daman

Scotland
The Amiga Center of Scotland — Edinborough

Sweden
Nicholas Pearson — Lund

Switzerland
Software Land AG — Zuerich

West Germany
Ralph Hollax — Huerth

Amazing Computing™ is also available in most B. Dalton Booksellers stores, B. Dalton Software stores, Crown Books, and Software Etc. locations.

Amazing Mail

Dear AC,
The article "Taking the Perfect Screen Shot" by Keith Conforti (in Vol 2 #10 of *Amazing Computing*) was informative for the most part. However, I would like to clarify a few details and possibly help readers to shoot even better pictures from the Amiga screen.

While using a zoom lens may be better than using the "normal" lens that comes with most cameras, zoom lenses have their own problems with distortion. These problems are minimized by using a smaller aperture than the f/4 setting recommended by Mr. Conforti. Usually setting your aperture for two stops smaller than the lens' largest aperture will give the best resolution for that lens. I use a 55mm macro lens. This is a lens corrected for close working distances and gives a fairly flat image. I would recommend a 100mm macro lens, but that is out of most peoples' price range. A zoom or telephoto lens at a medium aperture (f/8 or f/11 will fit most needs.

Taking a meter reading off the screen to determine exposure is also not a good idea. Without getting too deeply into the subject, most camera meters are designed to average all the light and dark areas of a scene and come up with an exposure that is the equivalent of what is called a "middle gray." That means if you have a generally light screen, the camera meter will underexpose the film to make it "middle gray"; if the screen is very dark, the meter will overexpose the film to get that gray. This is not good. I can see why Mr. Conforti recommends as many as 19 different exposures to get one good screen shot!

In reality, all you need do is find the one correct exposure for your Amiga's monitor screen. Once you have found that ideal exposure, you never need to change it again!

In the course of doing a set of title slides for my company, I made a set of test exposures using Kodak Ektachrome 200 slide film. As a test picture, I drew a gray scale from white to black and a set of color bars in red, green, blue, cyan, magenta and yellow.

My initial exposure was based on Kodak's recommendation of 1/2 second at f/8 for EI 200 film. Exposure for slide film is more critical than for print film. A half-stop of exposure either way can make a big difference. Therefore, I exposed a set of slides at 1/2 stop differences over a two stop range. This first set gave me a good exposure setting—but also revealed two problems.

First, the pictures were far too blue. The gray scale I used in my test picture was very bluish. I checked with my Kodak Photoguide and found that they recommend a Wratten CC40 Red filter when photographing color TV screens. Mistakenly, I bought a CC20 Red filter, but re-shot my test picture anyway. This time the grays were a very neutral gray and the other test colors came out better. This is because the excess blue from the monitor was not contaminating the other colors, too. I ordered a CC40 Red filter and will test it when I get it.

The second problem was that with proper exposure the film would not record all 16 steps of gray in the gray scale. The black and darkest gray bars were indistinguishable. This is understandable, since no slide film could record that exposure range. I have not explored yet whether this also happens with pure colors. When I do, I will report my findings to you. In the meantime, I would recommend not using the darkest gray near blacks in pictures you want slides of.

I also tested the effect of the monitor's Contrast Control on a picture shot off the screen. I set all controls at their mid-point settings and varied the Contrast Control setting while shooting my test picture. With the slides I shot, I could see no difference wherever the contrast control was set.

Anyway, my exposure for Ektachrome 200 film and the standard Amiga monitor (with all controls at their mid-point setting) is 1/2 second at f/8-11 (i.e. the half-stop between f/8 and f/11) using a CC20 Red filter. Kodak recommends a 1/8 second or longer exposure with focal-plane shutter cameras. As was stated in your article, this is to prevent the appearance of tv scan lines in the photograph.

This exposure should not change no matter what is on the screen. In my experience, I find this to be true. Of course, all my work has been done with slide film; however, print film should be easier because it is less sensitive to exposure variations. You can over- or under-expose print film by one or two stops and still get a decent picture.

In any case, this may not be the proper exposure for everyone's film/monitor combination. It will, however, be a good starting point. You can run your own tests starting with this exposure setting.

I hope I have helped out some of your readers who wish to take good quality shots off their monitors. I realize most of your readers are into programming rather than photography, so I hope I haven't gotten too technical. Thanks for your excellent magazine.

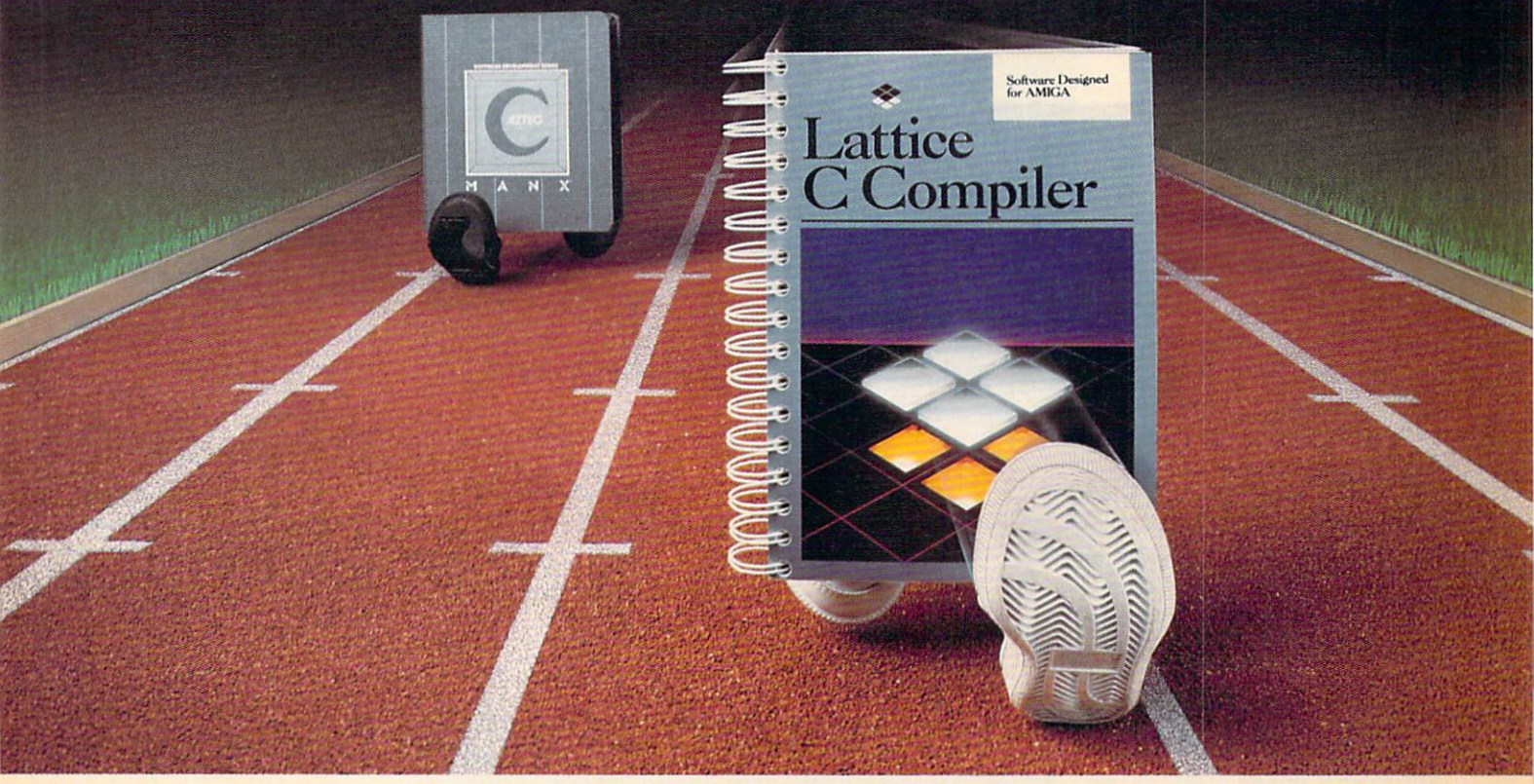
Sincerely Yours,
Marc D. St. Onge
Woronoco, MA

Thanks for some excellent suggestions! Our readers especially interested in slides will appreciate your tips. However, if the room is completely black and all luminescence is dependent upon screen light, different exposures will most likely be required. At least that is my experience when using through-the-lens metering.

Keith Conforti
Art Director
Amazing Computing™

•AC•

C who's winning the race. Lattice C for Amiga.



Lattice C has long been recognized as the best C compiler. And now our new version 4.0 for Amiga™ increases our lead past the competition even further.

Ready, set, go. The new Lattice AmigaDOS C Compiler gives you faster, more efficient code generation and support for 16 or 32-bit integers. There's direct, in-line interface to all Amiga ROM functions with parameters passed in registers. What's more, the assembler is fully compatible with Amiga assembler syntax.

More great strides. The linker, Blink, has been significantly enhanced and provides true overlay support and interactive recovery from undefined symbols. And you'll have a faster compile and link cycle with support for pre-linking.

	Lattice® Version 4.0	Manx® Version 3.40
Dhrystone	1294 Dhrystones/second	1010 Dhrystones/second
Float	22.20 Secs. (IEEE Format) 10.16 Secs. (FFP Format)	98.85 Secs. (IEEE Format) 17.60 Secs. (FFP Format)
Savage (IEEE)	47.67 Secs./.000000318 Accuracy	119.6 Secs./.000109 Accuracy

There's no contest. Standard benchmark studies show Lattice to be the superior C language development environment. With stats like these, it's no wonder that Commodore-Amiga has selected Lattice C as the official Amiga development language.

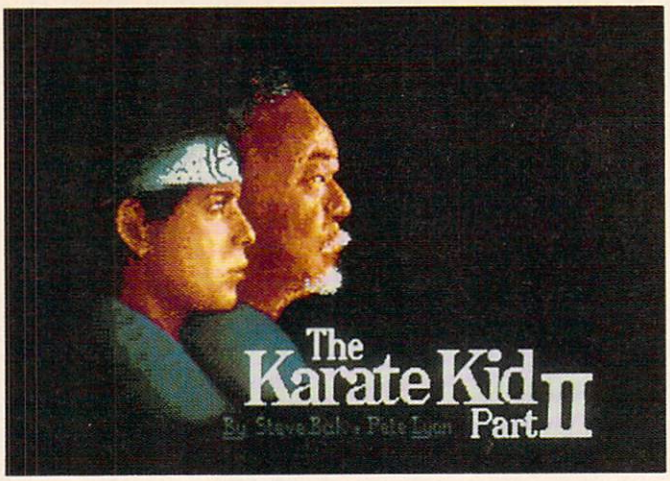
Going the distance. You'll experience unsurpassed power and flexibility when you choose from several cost-effective development packages. There is even a full range of supporting products, including a symbolic debugger, resource editor, utilities and specialized libraries.

You'll discover that your software purchase is backed by an excellent warranty and skilled technical support staff. You'll appreciate having access to LBBS—one of the world's first 9600 baud, 24-hour bulletin board services. And you'll be able to conference with other Lattice users through the Byte Information Exchange (BIX) network.

Cross the finish line. Order your copy of the Lattice AmigaDOS C Compiler today. We'll supply the speed. You bring the running shoes.

Lattice
Subsidiary of SAS Institute Inc.

Lattice, Incorporated
2500 S. Highland Avenue
Lombard, IL 60148
Phone: 800/533-3577
In Illinois: 312/916-1600



Karate Kid II is a fast action game based on the same-named movie. The game pits you against computer-controlled opponents in a series of Karate battles (Two-player mode is also available.). You are Daniel, and to win the game, you must battle 11 opponents, each more skillful than the last.

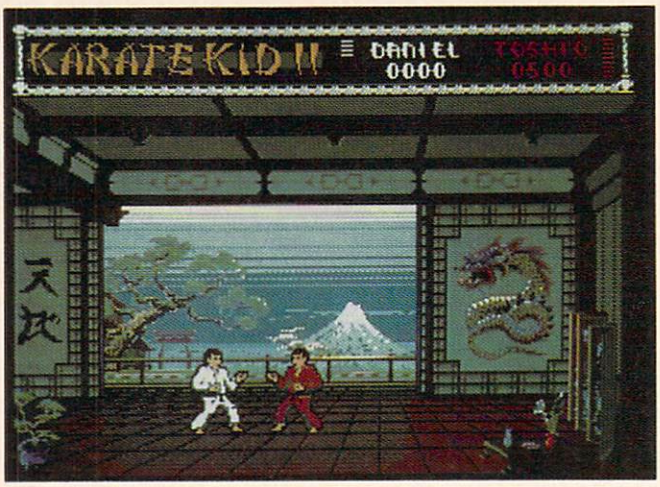
You control your character with a joystick plugged into the second mouse port (In the two-player game, you must remove the mouse from the first mouse port and plug the second joystick there.). Daniel moves, throws punches, and kicks depending on which direction the joystick is pointed. For example, to do a flying kick, you must press down the joystick button and point forward.

Game movement is made in accordance to the direction the character is facing. This perspective can be a bit confusing at first. A joystick command that works one way when Daniel faces towards the right side of the screen is translated to the opposite direction when he faces the left. To make Daniel throw a high punch when he faces right, you must point the joystick diagonally to the upper right. If he faces left, you must point the joystick towards the upper left. It's a bit awkward at first, but, after a few games, you become accustomed to the controls.

After every two screens of play, you get a chance to earn bonus points. The bonus screens are randomized, so they do not come up in any particular order.

Reviewed by
Stephen R. Pietrowicz

One bonus screen depicts Daniel's mentor, Miyagi, holding chopsticks poised to catch a fly. The fly moves around the screen, and you must use the joystick to catch the fly with the



chopsticks. The animation on this screen is amusing. Miyagi's eyes follow the fly, and the hand moves and closes the chopsticks when the joystick button is pressed.

I don't really like the screen. It's too randomized, and there is little skill involved. The area in which you can move Miyagi's hand is very small. Sometimes the fly stays in the "catchable" area for only a short time. It's impossible to catch the fly if it doesn't move into that small area.

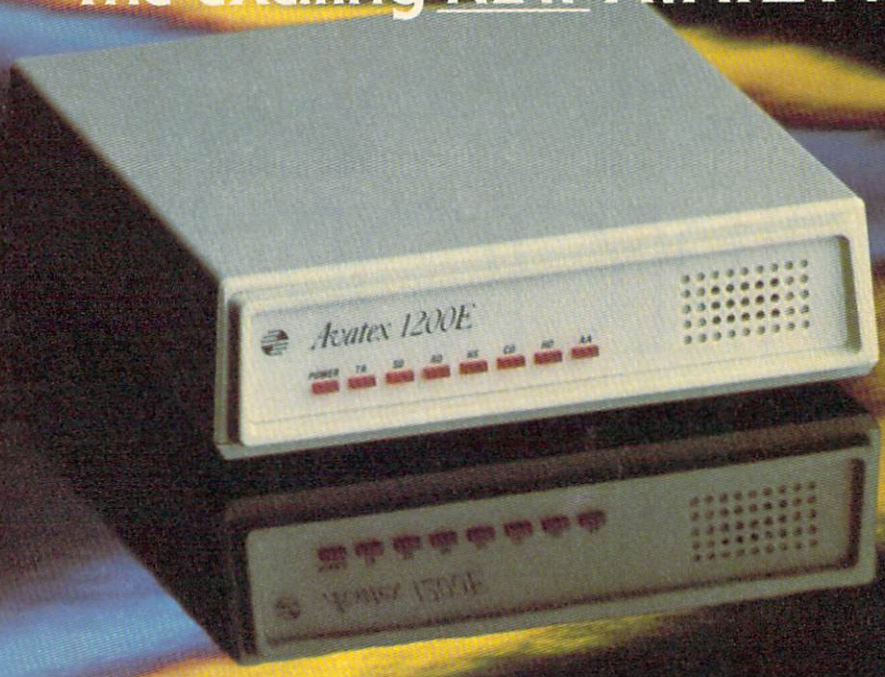
The other bonus screen depicts Daniel facing six sheets of ice, with a small drum in the upper corner. You must toggle the joystick handle left and right rapidly to simulate twisting the drum handle. The more quickly you toggle, the faster the beads hit the drum. Once you have the drum going as quickly as you can, you press the joystick button to start Daniel's swing, and then release the button for the follow-through. This skill requires some practice, but it's fairly easy to master smashing all 6 sheets of ice.

The problem with this screen is the tremendous flicker when Daniel hits the ice. It appears the animation wasn't double-buffered. Double-buffering draws objects in a separate area while another frame is displayed. Once the drawing is complete on that screen, it is displayed, and an object is drawn onto the screen. Double-buffering eliminates the flicker that occurs when all pictures are drawn on a single screen.

At the end of the game, a drum is displayed in the upper right hand corner of the screen. You must discover the secret of the drum. To tell you the truth, even though I've won all the battles in the game, I'm not quite sure about the secret of the drum. The game doesn't give you any hints to help you find out the secret either!

(continued on page 99)

GOOD THINGS COME IN SMALLER PACKAGES.
The exciting NEW AVATEX 1200 E.



$85
TOTAL DELIVERED PRICE.

AVATEX
Authorized Sales & Service

ACTUAL SIZE ONLY 5" X 6"

FREE! Communication Software & Compuserve Access Time with each MODEM.

We give you a great price...
+ TOLL FREE ASSISTANCE
+ IMMEDIATE DELIVERY
+ NO CREDIT CARD SURCHARGE
+ NO SERVICE OR HANDLING FEES

+

FREE SHIPPING!

TOTAL HAYES COMPATIBILITY
8 LED STATUS INDICATORS
AUTO DIAL AND ANSWER
TONE OR PULSE DIALING

NEW 1200 E FEATURES
CCITT COMPATIBILITY
CALL PROGRESS DETECTION
INTERNAL SPEAKER
2 YEAR WARRANTY

Thousands of customers have purchased the AVATEX 1200 from Megatronics. New technologies now make it possible to offer an improved version of this popular Modem.

The 1200 E is amazing.
It weighs less and is smaller.
It has even more features.
It's still the same low price.

MEGATRONICS

MEGATRONICS, INC. P.O. BOX 3660, LOGAN, UT 84321

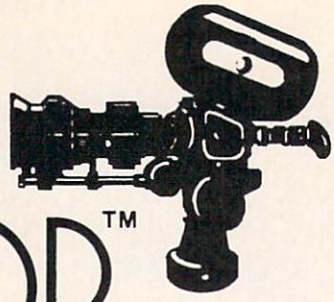
To Order
CALL FOR FREE CATALOGUE
CREDIT CARDS VERIFIED FOR YOUR PROTECTION

800-232-6342
INSIDE UTAH (801) 752-2642 FAX (801) 752-8752

We'll beat any advertised price.
MEGATRONICS, INC., P.O. BOX 3660, LOGAN, UTAH 84321

Dream on...

THE DIRECTOR™



Envision a creative freedom you've only *dreamed* about... building animations, presentations and computer films that use the *full* power of the Amiga. Imagine page flipping, color cycling, text generation, even Videoscape 3D™ animations, all *combined* at the same time on the same screen. Until now this has been just a dream. Now the dream comes true with the Director.

Combine pictures in any resolution from any paint program or digitizer you use with your Amiga. With the Director you write a script to display them in unlimited ways. You can page flip full or partial screens, use drawing commands, create special effects, generate multiple font text, execute AmigaDOS commands, and much more. This is only the beginning of the freedom you give yourself with the Director.

From the simplest slideshow to the most sophisticated desktop video, the Director will help turn your dreams into reality.

Professional display and animation software for the Amiga™

Other features include:

- Effects: Fades, Dissolves, Wipes, Blits, Stencils
- Digitized soundtrack module
- Preload images, fonts, and sounds up to your memory limit
- Supports HAM and overscan
- Keyboard and mouse interaction
- Random number generator
- Text string and file input and output
- No copy protection

And more...

$69.95

Check or money order payable to:
Right Answers

Plus $3 shipping and handling
Calif. residents add 6% sales tax

The Right Answers Group
Department C
Box 3699
Torrance, CA 90510
(213) 325-1311

Amiga is a trademark of Commodore-Amiga, Inc. Videoscape 3D is a trademark of Aegis Development, Inc.

GO-64!

Commodore 64 Emulator

by John Foust, Rick Wirch, and James O'Keane

After the dismal performance of the Transformer IBM PC software emulator, Amiga owners should be wary of software emulators. The GO-64! Commodore 64 emulator from Software Insight Systems (SIS) is no exception. The intent of GO-64! is admirable. After all, if the hordes that now own Commodore 64s can be drawn to the Amiga product line, the Amiga will be a very popular computer.

When people started using the Transformer, they were disappointed. It worked slowly, and only a minimal subset of the IBM PC computer was emulated. Too many people believed the marketing hype of 100 percent emulation at 80 to 100 percent speed. Too many people thought less of Commodore because of the misleading statements.

Unfortunately, the same things can be said of this incarnation of the Commodore 64 emulator. First, emulated Commodore 64 programs run slowly. The Commodore 64 was no speed demon to start with, and GO-64! is certainly no faster. A simple program written with a compiled BASIC appeared to run at about 30 to 50 percent of normal speed. An arcade game, written in pure assembly language, was tremendously slow. It ran at about 10 percent of its regular speed, even with the recommended settings from SIS.

The emulation seems inaccurate in spots, too. Updates of the screen take place at a much slower rate than the program is running, so things that should be happening simultaneously instead happen in sequence. For instance, a series of sprites on the screen in one program should have danced and changed color in synch, but did not do so.

The GO-64! package includes a small interface that connects to the parallel port. This interface provides a link to the DIN connector of Commodore peripherals, such as the 1541 disk drive. The GO-64! parallel port interface is the proper sex for the Amiga 500 and 2000, but not for the Amiga 1000. If you have an Amiga 1000, you must buy their special adapter. An ordinary gender-changing adapter will not work because of changes made in the pin assignment on the Amiga 500 and 2000 parallel ports.

ROM Images

An important part of a Commodore 64 is the operating system memory, or ROM. This memory is stored in circuit chips inside the Commodore 64, similar to the way the Kickstart is stored in an Amiga 500 or 2000.

GO-64! cannot supply a copy of the Commodore 64 ROMs on disk because Commodore owns the programs in the ROM chip. They negotiated unsuccessfully with Commodore in an attempt to supply a copy with the product.

A nine-line BASIC program is supplied in a NotePad file. You must enter this program on the Commodore 64. When run, the program creates two files on the Commodore 64 disk. These files contain a copy of the system ROMs. These files must be moved from the Commodore 64 disk to an Amiga disk.

How can you do this? You could use a program such as Central Coast Software's Disk-2-Disk to transfer the Commodore 64 files to an Amiga disk, if you have a model 1020 5 1/4 inch disk drive. You could use a terminal program and modem to call another computer system and send the files there. You could then retrieve them with an Amiga and save them to Amiga format. You also could connect the two computers with a null modem cable and use telecommunications software to move the files.

The nine-line Commodore 64 program should have been supplied on a Commodore 64 disk, instead of the Amiga disk. A public domain or

(continued)

custom terminal program of some kind should have been provided on disk. Better still, SIS could have supplied automated, custom programs for both computers to ease the ROM image transfer process. They could then sell the null modem cable needed for the transfer, too.

A NotePad file contains several tips and corrections to the manual. It also contains the short BASIC program mentioned above. Something is wrong with the file on the disk, however—the NotePad program will not display the entire file. It stops at a certain page. Using the "type" command does not work because of the strange characters in the file. The final characters in the file erase the screen. Instead, this file must be sent to a printer or cautiously read using the "type" command and some other key to pause the listing.

Known Bugs

Many times, the GO-64! program would Guru as it was booting. According to SIS, this is a known bug—the program does not work with expansion memory, on any Amiga model. You can run the NoFastMem program that comes with the Amiga 500 to remove the expansion memory from the system; then the program will work. The GO-64! disk includes a GO-64! Preferences program. Like the GO-64! program, it doesn't work with expansion memory. The imagery on the gadgets is garbaged unless the NoFastMem program is run.

> "Although it is an amazing piece of programming, it is far too slow and cantankerous to use on a regular basis."

There is another known bug in the first version of the program. The German version of the Amiga 2000, as well as some of the early units from West Chester, have slightly different keyboard controllers. While this does not affect most software, it does trip up programs that access the keyboard controller directly. GO-64! is one of these programs that accesses the hardware directly.

Copy-protection

The GO-64! disk is heavily copy-protected. The manual warns that your floppy drive may be damaged if you try to copy the disk. This situation is hardly possible, so why do they make the threat?

The immediate need for copy protection is not apparent. The company's fear of Commodore 64 pirates using their product to continue pirating on the Amiga is justified. At this time, though, the GO-64! hardware is necessary for using the emulator—without it, there is no way to access mass storage. Sure, you could program in BASIC all day, but you wouldn't be able to save your work. There is a chance that the GO-64! hardware may be easy to duplicate, but that idea is somewhat far-fetched and probably would not occur on a large scale.

Mouse Troubles

The mouse must be removed from the first joystick port after the emulator starts. The Commodore 64 acted somewhat strangely—if the joystick was in the game port, strange characters appeared on the screen as it was moved. However, this reaction goes against the advice of a small sticker on the Amiga 500 that warns, "To prevent damage turn power off before connecting or removing cables."

The second mouse port is directly connected to one of the custom chips. There is potential for damage here. The hood of the Amiga 500 mouse is metal, so there is a possibility that careless insertion could short and damage the computer (not that this potential deters many Amiga owners from plugging things anytime).

Turbo Disks

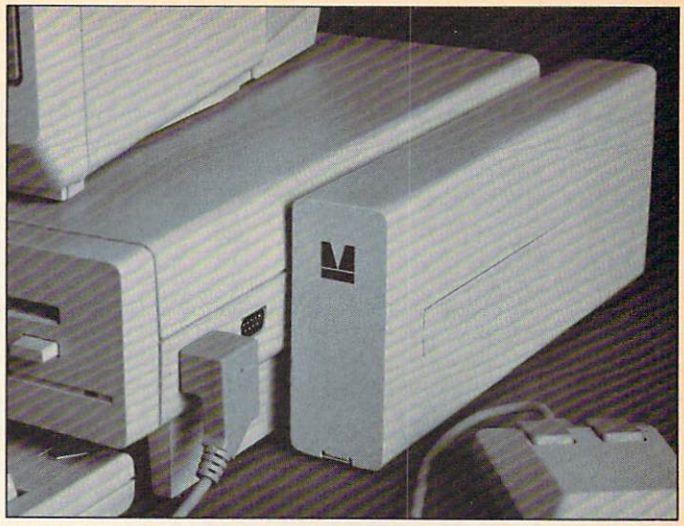
To paraphrase the words of a former Amiga Los Gatos employee, "The 1541 disk drive was one of the best computers made by Commodore." It is

(continued on page 14)

AVAILABLE NOW!

StarBoard2

Auto-Configuring
Fast RAM
Zero Wait States
User Expandable
from 512k to
2 Megabytes
Bus Pass-Through
MultiFunction
Option: battery/
clock, FPU,
parity, Sticky-Disk



If you've owned your Amiga® for a while now, you *know* you definitely need more than 512k of memory. You probably need *at least* double that amount...but you might need as much as an additional two megabytes. We want to urge you to use **StarBoard2** as the solution to your memory expansion problem –and to some of your other Amiga-expansion needs as well!

It's small, but it's BIG–

Since most of you want to expand your Amiga's memory without having to also expand your computer table, we designed **StarBoard2** and its two optional "daughterboards" to fit into a sleek, unobtrusive Amiga-styled case that snugly fastens to your computer with two precision-machined jackscrews.

The sculpted steel case of **StarBoard2** measures only 1.6" wide by 4.3" high by 10.2" long. You can access the inside of the case by removing just two small screws on the bottom and pulling it apart. We make **StarBoard2** easy to get into so that you or your dealer can expand it by installing up to one megabyte of RAM on the standard **StarBoard2** or up to two megabytes by adding in an Upper Deck.

This card has decks!

The basic **StarBoard2** starts out as a one megabyte memory space with 0k, 512k, or one megabyte installed. If you add in an optional **Upper Deck** (which plugs onto the Main Board inside the case) you bring **StarBoard2** up to its full two megabyte potential. You can buy your **StarBoard2** with the Upper Deck (populated or unpopulated) or buy the Upper Deck later as your need for memory grows.

And you can add other functions to **StarBoard2** by plugging in its second optional deck –the Multifunction Module!

StarBoard2: functions five!

If we count Fast Memory as one function, the addition of the **MultiFunction Module** brings the total up to five!

THE CLOCK FUNCTION:

Whenever you boot your Amiga you have to tell it what time it is! Add a MultiFunction Module to your **StarBoard2** and you can hand that tedious task to the battery-backed, real-time clock/calendar. A small piece of MicroBotics software in your WorkBench Startup-Sequence reads the clock and automatically sets the time and date in your Amiga. And the battery *is* included (we designed it to use an inexpensive, standard AAA battery which will last at least two years before needing replacement).

THE FLOATING POINT FUNCTION:

If any one aspect most characterizes the Amiga it's *fast* graphics! Most graphic routines make heavy use of the Amiga Floating Point Library. Replacing this library with the one we give you with your MultiFunction Module and installing a separately purchased Motorola 68881 FPU chip in the socket provided by the Module will speed up these math operations from 5 to 40 times! And if you write your own software, you can directly address this chip for increased speed in integer arithmetic operations in addition to floating point math.

THE PARITY CHECKING FUNCTION:

If you install an additional ninth RAM chip for every eight in your **StarBoard2**, then you can enable *parity checking*. Parity checking will alert you (with a bus-error message) in the event of any data corruption in **StarBoard2**'s memory space. So what good is it to know that your data's messed up if the hardware can't fix it for you? It will warn you against saving that data to disk and possibly destroying your database or your massive spreadsheet. The more memory you have in your system the more likely it is, statistically, that random errors will occur. Parity checking gives you some protection from this threat to your data residing in Fast RAM. Note that the Amiga's "chip" RAM cannot be parity checked.

THE IMMORTAL MEMORY DISK FUNCTION (STICKY-DISK):

When you've got a lot of RAM, you can make nice big RAM-Disks and speed up your Amiga's operations a lot! But there's one bad thing about RAM-Disks: they go away when you re-boot your machine. Sticky-Disk solves that problem for you. It turns all of the memory space inside a single **StarBoard2** into a Memory Disk that will survive a warm-reboot! When your Amiga attempts to grab a **StarBoard2** in Sticky-Disk mode, a hardware signal prevents the system from acquiring the **StarBoard2** as FastRAM (and thereby erasing your files) –instead it is re-recognized as a Memory Disk and its contents are preserved intact. If you want to work rapidly with large files of data that are being constantly updated (such as when developing software) you can appreciate the Sticky-Disk!

Fast RAM –no waiting!

StarBoard2 is a *totally* engineered product. It is a ZERO WAIT-STATE design, auto-configuring under AmigaDOS 1.2 as Fast RAM. Since AmigaDOS 1.1 doesn't support autoconfiguration, we also give you the software to configure memory in 1.1.

Any applications software which "looks" for Fast RAM will "find" **StarBoard2.** And you'll find that your applications run more efficiently due to **StarBoard2** on the bus.

A passing bus? Indeed!

What good is an Expansion Bus if it hits a dead end, as with some memory cards? Not much, we think –that's why we carefully and compatibly passed through the bus so you could attach other devices onto your Amiga (including another **StarBoard2**, of course!).

The sum of the parts...

A really nice feature of the **StarBoard2** system is that you can buy exactly what you need now without closing off your options for future exapansion. You can even buy a 0k **StarBoard2** (with a one megabyte capacity) and populate it with your own RAM (commonly available 256k by 1 by 150ns memory chips). When you add **StarBoard2** to your Amiga you have a powerful hardware combination, superior to any single-user micro on the market. See your Authorized Amiga Dealer today and ask for **StarBoard2**

SUGGESTED RETAIL PRICING:	
StarBoard2, 0k (1 meg space):	$349
StarBoard2, 0k (2 meg space):	$395
StarBoard2, 512k (1 meg space):	$495
StarBoard2, 1 meg (1 meg space)	$595
StarBoard2, 2 megs installed:	$879
StarBoard2, 2 megs & MultiFunction:	$959
Upper Deck, 0k (1 meg space):	$ 99
MultiFunction Module:	$ 99
also available:	
Standard 256k memory card:	$129
MAS-Drive20, 20 meg harddisk:	$1495
MouseTime, mouseport clock:	$ 50

MicroBotics, Inc.
811 Alpha Drive, Suite 335, Richardson, Texas 75081 / (214) 437-5330

tremendously slow, however. How slow? Its speed is comparable to a 2400 baud modem, about 200 characters per second.

After several years, experienced Commodore 64 programmers developed methods of increasing the transfer speed of the 1541 drive to almost acceptable levels. These methods were called "turbo loaders." Most of these techniques involved massive replacement of the Commodore 64 operating system software. Some of the turbo load implementations were in loadable software; others used ROM cartridges.

In the Commodore 64, these turbo load programs are very fragile. They contain many time-critical sections of code. If the interaction between hardware and software is not perfect, it will not work. In the case of the GO-64! emulator, the hardware (such as the 1541 disk drive) is still working at full speed, but the software is not keeping up. Software Insight Systems has developed a method called Hyper-Code to get around these incompatibilities.

SIS claims they will have free Hyper-Code modules to replace the turbo load code for popular Commodore 64 programs, similar to the way the "brain" files work with the Marauder II copy program.

Other strange things happen on the 64. Some programs actually send program code to the 1541 drive to be executed there by the microprocessor that controls the drive.

Hyper-Code

According to SIS, the Hyper-Code files improve the performance of any Commodore 64 program. They are working with companies such as Activision, Electronic Arts, and Epyx in order to work more reliably with these companies' products. They say these companies have been very helpful so far.

The GO-64! disk contains a Hyper-Code file for Berkeley Software's GEOS operating system, version 1.2. This file is a windowed operating system for the Commodore 64. An update to GO-64! will have the Hyper-Code for GEOS version 1.3. The Hyper-Code files are freely distributable. They will be available on the SIS tech support bulletin board, as well as on the commercial networks.

At this time, there is no support for Amiga file storage; you must use a 1541 disk drive. SIS reports that they are currently working on a file transfer program for moving files from Commodore 64 format to Amiga format. They also hope to support the 1581 disk format on regular Amiga 3 1/2 drives. The 1581 is a new 3 1/2 drive for the Commodore 64 that holds 720K per disk.

Amazing Software

As programmers, we must praise the programmers at Software Insight Systems for what they have created so far, even if it is slow and inaccurate. A software emulator of a computer as complex as the Commodore 64 is a tremendous programming feat. We understand the magnitude of the program. The Amiga Transformer is a similarly astounding feat. While a program may be admirable from a programming standpoint, it may still be useless from a user standpoint.

Summary

At this time, we cannot recommend the GO-64! emulator. Although it is an amazing piece of programming, it is far too slow and cantankerous to use on a regular basis. It may be improved in the future, but without an increase in speed, it is little more than a curiosity. We get the strong impression that the first version of GO-64! was rushed to market. The foul-up in the NotePad file is evidence.

•AC•

Software Insight Systems
16 East International Drive
East Granby, CT 06026
(203) 653-4589 $69.95

About the Authors

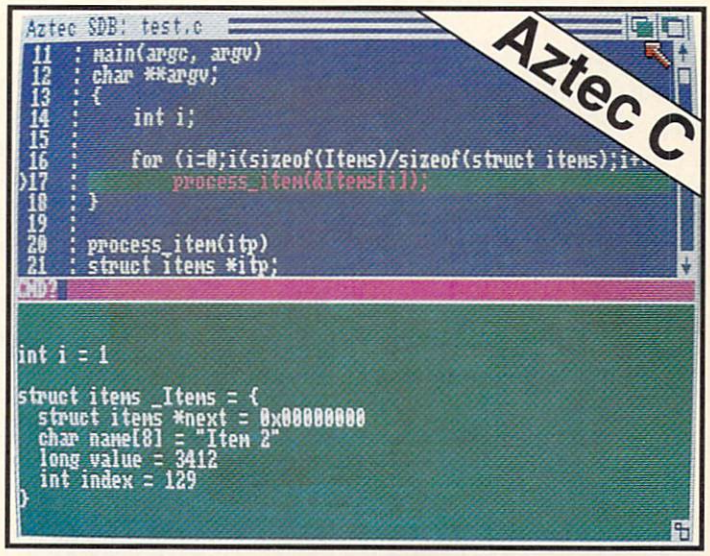
John Foust, Rick Wirch, and James O'Keane were once programming wizards on the Commodore 64, but now work on the Amiga. They were programmers at Sight & Sound Music Software, the makers of popular Commodore 64 software including Music Video Kit, Kawasaki Synthesizer and Rhythm Rocker, and the Incredible Music Keyboard.

SOURCE LEVEL DEBUGGER

Announcing the Manx Aztec C Source Level Debugger for the Amiga!

NOW THE MOST ADVANCED COMPUTER IN THE WORLD HAS THE MOST ADVANCED SOURCE LEVEL DEBUGGER IN THE WORLD:

Manx Aztec *SDB.*



- View your C Source
- Enter commands
- View your command output ...

... all at the same time!

Save Time and Effort

If you're a pro at working with low level Amiga debuggers, you'll recognize what a source level debugger can mean—*time savings.* Time you'd rather spend *creating* than debugging. And if you're a beginner, *SDB* will make you a pro in no time.

Outstanding Features

That's why our new windowed *SDB* is so spectacular—because it's full of exciting features that make debugging a breeze. Of course, *SDB* has all of the features you expect from a debugger—like line-by-line tracing. Conditional breakpoints on lines, functions, or variables. Examination, modification, and display of global, local, and static variables, structures or expressions by name.

But *SDB* is also full of unexpected, incredibly sophisticated features. There's reusable command macros and procedures. Back tracing. Active frame context switching—just to name a few. Wait till you see *SDB* in action—it will blow you away!

Our Commitment to You

Manx Software Systems is the leading edge in C development systems. That means continual updates that bring the best to you.

Experience version 3.6 of Aztec C68k/Am today. And enjoy the most advanced debugger available ... on *any* computer.

Standard Features of Aztec C68k/Am 3.6:

- optimized C with selectable 68020 and 68881 support
- 680x0 Macro Assembler with 68881 support
- linker/librarian with overlays, scatter load, and segmentation
- symbolic debugger
- supports Amiga object format
- UNIX, AMIGA, and general purpose run time routines
- runs under CLI and supports all Workbench functions
- creates CLI and Workbench applications
- Supports both 1.1 and 1.2 Amiga DOS
- 600 pages of documentation and great example programs

Extended Features of Aztec C68k/Am 3.6

- UNIX utilities make, diff, grep, obj, ord, and vi
- special math support libraries for 68881 and Manx IEEE emulation

Portability: Aztec C is also available for the Macintosh, Apple II, MS-DOS/PC-DOS, CP/M-86, TRS-80, ROM, and others. Aztec C68k Third Party Software: A large array of support software is available for Aztec C86k. Call or write for information. The following is a list of the most requested products: **Power Windows • Amiga View • Key to C • Amiga Lint • Metascope.** Immediate Delivery Available for Most Destinations. Aztec C is available on a thirty day money-back guarantee. **C.O.D., VISA, American Express, MasterCard, wire (domestic and international),** and terms are available.

AZTEC C VERSIONS TO SUIT YOUR NEEDS:

Aztec C68K/Am-p Professional System $199
Includes all of the Standard Features

Aztec C68K/Am-d Developer System $299
Includes all of the Standard and Extended Features

Source Level Debugger. $75

Library Source $300

Order Now At No Risk Or Call For Information

1-800-221-0440

In NJ or outside the USA call: 201-542-2121
Telex: 4995812MANX Fax: 201-542-8386

MANX

Manx Software Systems • One Industrial Way • Eatontown, NJ 07724

SEE FOR YOURSELF WHAT ALL THE FUSS IS ABOUT ...

Order our Manx Aztec C68k/Am with 30-day satisfaction guarantee. We're convinced that once you see *SDB* at work, no other debugger will ever be good enough again. But if you don't believe us—try us! We're offering an SDB Demonstration Disk for just $5. Simply call 1-800-221-0440 (NJ call 201-542-2121) and order your copy of Aztec C or your Demo Disk today.

A-Talk-Plus

Chatting with Mainframes!

by Brendan Larson

Several quality public domain terminal packages are available for the Amiga. Many such programs do credible jobs of providing the Amiga community with necessary telecommunication functions, such as XModem, ASCII Text, and Binary transfers. All of these can be used for transferring a menagerie of public domain programs over the phone lines.

What about the Amiga user who needs to communicate with a mainframe? Several terminal programs available for the Amiga can't handle this task (since the average Amiga user doesn't need this feature). If you're looking for a full-fledged terminal program, packed with many features including those that access mainframes, you have an alternative—A-Talk-Plus from Felsina Software!

Marco Papa is the author of A-Talk-Plus (ATP). I have tested ATP extensively on-line with a variety of computers in different emulation modes, ranging from Amiga BBSs like Aliens and Casa Mi Amiga, to mainframes like Vax 11/750s and Vax 8200s at a large meteorological data base called Accu-Weather (See *Amazing Computing* V2.7 for more on Amiga Weather Graphics systems.). I have also successfully downloaded and uploaded text and binary files to an IBM 3081 (using the Conversational Monitor System (CMS) and an IBM 7171 communications field which converts Amiga ASCII to IBM EBCIDIC via A-Talk-Plus's Kermit and VT-100 emulation mode).

The main difference between ATP and its predecessor, A-Talk, is the Tektronix 4010/4014 graphics emulator. Tektronix emulation, usually run on mainframes or minis, quickly transfers high resolution CAD/CAM images (in the form of controlled ASCII characters) to the end user. I used the Tektronix graphics emulator to dial the Accu-Weather data base, connect to their Vax mainframe, and obtain high resolution pictures of meteorological weather data (see Figure 1) to help me prepare for a weather forecast.

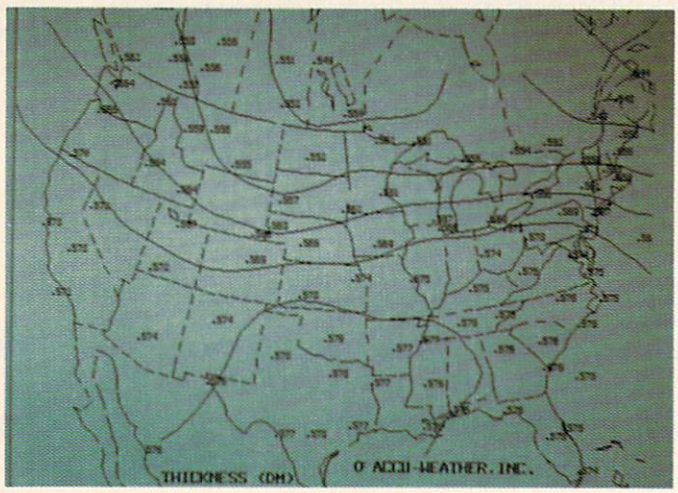


Figure One *Courtesy of Accu-Weather, Inc.*

What's really incredible about the Tektronix graphics emulator is that it to operates in the Amiga's full-video (overscan) mode. On my Sony KV-1311CR monitor, I was able to get an approximate total screen resolution of 704 X 460, although you can set the parameters to 704 X 480. Working in overscan is useful when you are using the Tektronix graphics emulator because much more data can be displayed.

Another powerful feature of ATP is its ability to Zoom in the hi-res image. Unlike the zooming done in DeluxePaint 2, where pixels are exploded, Papa has written a special zoom feature that creates a superbitmap of 1024 X 1008 (a large virtual screen) which can be panned for a detailed look at various locations using scroll bars on the Zoom window. This zooming capability is great; during my experiments, I analyzed weather data (such as Figure 1) on a much smaller scale, with better resolution. The image in the Zoom Window is similar to viewing a Quadrant of the whole image on a grid (similar to the Quadrants in the Cartesian Coordinate System).

(continued on page 18)

Real time video and music.

SunRize introduces PERFECT VISION and STUDIO MAGIC to enhance the look and the sound of your Amiga.

Actual Unretouched Photos

STUDIO MAGIC is the ultimate music and sound workshop with features superior to editors selling for thousands more. Compatible with model 1000, 500 and 2000 Amigas. Input sounds from stereo, VCR or microphone (with PERFECT SOUND interface) or a keyboard (with MIDI interface).

Create flanges, delays, echoes, compression and expansion to speed up (or slow down) without pitch change, do backward masking, etc. Other features include comb filter, DC bias, AM, Fast Fourier Transforms and visual representation of the sounds in the buffer.

A dozen menu driven tools allow you to make a child's voice sound like a titan or turn a TV pitchman into an alien from another galaxy. Record MIDI input in real time. Mix voices and instruments and control their playback using a MIDI keyboard. Overdub from 16 digitized "sections" and play back four of them at a time. Supports advanced MIDI features such as tempo adjust and external sync. Store sounds in IFF "instrument" or "one-shot" (8SVX) files for use with other compatible programs.

SUGGESTED RETAIL PRICE: $99.95

PERFECT VISION is the state of the art, real time video digitizer for use with model 1000, 500 and 2000 Amigas. Input from a color (or black and white) video camera or a VCR. Perfect Vision will digitize the image, display it in 4096 colors (camera input only), then store it as IFF for later use in compatible programs. Captures an image in 1/60 of a second — 600 times faster than the competition. Supports 320x200 and 320x400 HAM and 16-color modes.

SUGGESTED RETAIL PRICE: $219.95

Available from Amiga dealers across America. For product information and support call:

SunRize Industries
3801 Old College Road
Bryan, TX 77801
(409) 846-1311.

(continued from page 16)

Because of the way the Tektronix Emulator works, all hi-res data plotted on the screen comes into the Amiga as ASCII control characters. ATP gives you the option of "capturing" these control codes in a capture buffer while on-line with the mainframe. After the captured text is stored, the image can be "Replayed" later. This saves the time and money spent on the system host, along with disk space (since an ASCII file requires less storage space than an IFF bitmap).

The Replay option puts the image on screen at an incredible rate of 19.2K baud from RAM: or 9600 baud from floppy disk. From there, the picture can be saved in either an Aegis Draw "layer" format or an IFF bitmap. The saved images can range from 1 to 4 bitplanes, depending on the number of colors you select (A mini "Tektronix Preferences" is provided, allowing for color selection.). On a 512K Amiga, only one bit plane is allowed. However, such an apparent color "shortage" is usually not a problem since most Tektronix 4014 images are monochromatic.

All images can be dumped to a printer as well. Finally, A-Talk-Plus makes it possible to use a large cross-hair (cursor) while interacting with the mainframe (required by certain Tektronix applications).

As for the rest of A-Talk-Plus, there are many more added features that should make the serious Amiga user smile! It is possible to emulate the following: VT-100, VTM4, VT-52, H19, Kermit, VT-640, ANSI, TTY, and Talk mode. In VT-100 emulation mode, extra fonts can be displayed, and the VTM4 mode (via VT-100) allows you to obtain colored text. According to the manual, A-Talk-Plus is also capable of accessing UNIX-based systems, but I did not have the opportunity to test ATP in such an environment.

One of the key features of ATP's Kermit emulator is its capability to Send and Receive to and from mainframes at both 8 data bits/no parity and 7 data bits/even parity! According to reliable sources, ATP is the only Amiga communications package at the time of this writing that can perform Kermit transfers in both 7 and 8 bit prefixes.

If you need to download or upload programs from a BBS, ATP supports ASCII Text Transfers, ARC files, and XModem-CRC binary file checking, as well as the all familiar Checksum routines. ATP's best feature—which I have not seen available in any other terminal package for the Amiga—is that it keeps track of the number of "retries," or attempts to send or receive data blocks. For instance, if the phone lines are noisy and a bad block of data is transmitted, ATP will try again (several times if necessary) to get a clean block of data! ATP also calculates the estimated upload time for files sent from the Amiga to another computer by taking into account the file size and the baud rate.

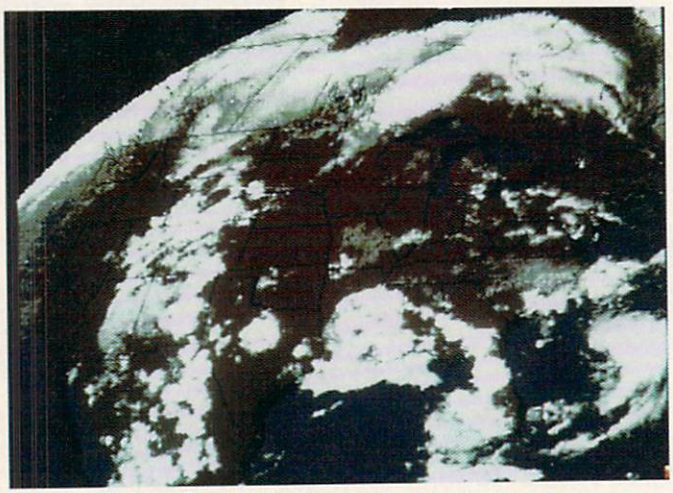


Figure Two *Courtesy of Accu-Weather, Inc.*

If you frequently log onto the same system in the same day, you can prepare an auto "Login Script" which actually waits for the other computer to send messages. Here is a scenario of an On-Line Login using an auto Login Script:

```
>Connect 1200
>Welcome to the Bunky BBS!
>
>Your Name: Brendan Larson
>
>Enter Password: Doodad
```

With the proper Login script, ATP waits for the phrase or prompt stating "Your Name:" and automatically replies with the necessary information, like "Brendan Larson." Login Script is programmed with ATP's very unique command code, explained thoroughly in the manual.

All ten Function keys on the Amiga, as well as ten extra Function or "macro" keys (a combination of the Shift and Function keys), may be defined, totaling 20 predefined functions. This feature may not be new to Amiga telecommunications specialists.

ATP can hold a large phone directory of the most commonly accessed BBSs or data bases, and will store all the function keys and settings for each. Another element that makes ATP a worthwhile investment is that it works with almost any Amiga compatible modem, even if it is not Hayes compatible!

Finally, another cute feature of A-Talk-Plus (probably where its name is derived from) is its ability to turn on the Amiga's built-in speech synthesis. ATP can read all incoming text data, or just the system commands. Of course, all ASCII control codes are filtered out, so only "real" words are heard.

(continued on page 20)

Other Products From The Other Guys	
Reason	$395.00
Omega File	$79.99
Promise	$49.99
KEEP-Trak GL	$49.99
AMT (Amortization Program)	$39.99
Match-It	$39.99
Math-A-Magician	$39.99
Talking Story Book (Christmas Stories)	$39.99
Musicial Slide Show Demo	$ 5.00

Call or write for more information.

SYNTHIA *High Performance Digital Synthesizer*

A state of the art music tool which will:
Create digital **IFF Instruments** for use with nearly all music programs!
Modifying existing **IFF Instruments.** Use *SYNTHIA* on digitized samples to add reverb, wow, and other enhancements.

SOMETHING FOR EVERYONE:

Additive Synthesis - a traditional method which can create almost any type of instrument.
Plucked String Synthesis - simulates plucked strings . . . right down to the 'pluck'.
Interpolative Synthesis - a method which introduces the natural imperfections found in instruments. (Instruments such as brass, woodwinds, pianos, etc.)
Percussion - build your own drum set . . . create any drum you desire.
Subtractive Synthesis - a simple method of creating instruments.
Special Effects - includes filtering, amplification, phasing, waveshaping, amplitude modulation, real reverb, and . . .
IFF Music Player - powerful and compact. Now you can enjoy those songs that needed a memory expansion before! Up to 32 tracks and 32 IFF Instruments! Supports chords, ties, etc.

NOW SHIPPING

IS IT LIVE . . . OR IS IT SYNTHIA?

Synthia uses the latest technology to generate realistic sounding instruments and even the new families of instruments sound real. A real synthesizer on a real computer!
Why buy digitized instruments when you can *SYNTHIAsize* them?
Requires AMIGA 512K

$99.99

THE OTHER GUYS
55 North Main Street
Suite 301-D
PO Box H
Logan Utah 84321

(801) 753-7620
(800) 942-9402

VISA MasterCard

(continued from page 18)

Papa has written an extended version of A-Talk-Plus called Digi Weather that allows you to access a real time meteorological data base, Accu-Weather. Digi Weather makes use of the same features as A-Talk-Plus, with the added capability of downloading GOES Satellite cloud pictures, as well as a collection of many other types of weather art designed by Accu-Weather artists and meteorologists.

While on-line with Accu-Weather, a meteorologist can take advantage of Digi Weather's Tektronix 4014 emulator by receiving hi-res "behind-the-scenes" weather data in the form of maps displaying isobars, etc. A meteorologist can also obtain a hi-res overscan (704 X 480) satellite cloud picture of the U.S. At 1200 baud, it takes approximately 13 minutes to obtain one of these maps. If, for instance, you want only a close-up shot of the Southwestern quadrant of the U.S., you could opt to download that as well, in a lo-res overscan mode (352 X 240). The latter, of course, saves time and money.

The good news for amateur meteorologists is that Digi Weather will be relatively inexpensive, and Accu-Weather, Inc. has special dial-up rates for the hobbyist (At this time, prices are not available.). Digi Weather is the first and only telecommunications software that allows the Amiga to receive satellite cloud pictures of this type (see Figure 2).

In conclusion, if you need a communications package that "does it all," or if you require access to a mainframe, I highly recommend A-Talk. If you need Tektronix 4010/4014 capabilities, A-Talk-Plus has a head start in the Amiga community. If you're a weather enthusiast like myself, Digi Weather combines all the features of A-Talk and A-Talk-Plus.

A-Talk-Plus $99.95
A-Talk $49.95

Marco Papa
Felsina Software
3175 S. Hoover Street, #275
Los Angeles, California 90007
(213) 747-8498

For more information about the Accu-Weather data base, call Accu-Weather, Inc. at (814) 237-0309 or write them at:

619 West College Avenue
State College, PA 16801

If you have any further questions, contact me at:

Brendan Larson
Channel 41 News Weather Department
7921-D Knottingham Circle
Darien, Illinois 60559
(312) 810-0304 or (312) 257-2818

•AC•

SYMPHONY MUSIC VIDEO

SYMPHONY MUSIC VIDEO continuously displays pictures and plays music. At the end of a selection, another picture and music piece is loaded. The music and pictures are all in IFF format so you can modify ours with your favorite Paint or Music program or use your own music and pictures and create your own MUSIC VIDEO. You can even have your MIDI synthesizer play the music.

The MUSIC VIDEO that is included is perfect for the Christmas Season. Traditional Christmas scenes and music are continuously played. Fun to listen to and watch. Adds to the best of seasons. **$24.95**

Symphony Songs

Ready to Play Music For

Deluxe Music, Music Studio, Sonix

A library of nearly 1,000 music masterpieces ready to play with your favorite music program. All selections are in both IFF and MUSIC STUDIO format. Space does not allow listing all songs in each volume, however, a few titles, the number of songs, and the total playing time is given.
Complete list of songs **$3.95.**
Each volume is **$24.95** and includes the complete list.

BEATLES Part 1
Vol 15 (21 Pieces 40 Min)
Let It Be, Yesterday, Eleanor Rigby, When I'm 64, . . .

BEATLES Part 2
Vol 40 (15 Pieces 40 Min)
Magical Mystery Tour, Lucy In The Sky With Diamonds, Penny Lane, . . .

CLASSICAL Part 1
Vol 27 (18 Pieces 40 Min)
Prelude #1, Moonlight Sonata 1st and 2nd Movement, . . .

CLASSICAL Part 2
Vol 34 (13 Pieces 40 Min)
Sonata In C Major, Jesus Joy Of Man's Desire, . . .

CLASSICAL Part 3
Vol 31 (14 Pieces 35 Min)
1st Piano Concerto, Polonaise Sonata In C Major, Etude #3, . . .

CLASSICAL Part 4 (Bach)
Vol 35 (22 pieces 30 Min)
Two Part Invention #1, Three Part Invention #6, Prelude and Fugue 1, . . .

CLASSICAL Part 5 (Bach/Clementi)
Vol 46 (24 Pieces 50 Min)
Choral #1, Sonata #1, Theme and 11 Variations From The 2nd Sonata, . . .

BEETHOVEN, BROADWAY, & BLUES
Vol 38 (15 Pieces 40 Min)
2nd Movement Of the Pathetique Sonata, Minuet In G, Fuer Elise, . . .

COUNTRY CLASSICS Part 1
Vol 41 (15 Pieces 45 Min)
Thank God I'm a Country Boy, Act Naturally, . . .

ROCK Part 1
Vol 32 (19 Pieces 50 Min)
AXEL F, Eye Of The Tiger, Both Sides Now, . . .

ROCK Part 2
Vol 16 (20 Pieces 40 Min)
Georgy Girl, Guantanamera, Theme From "Love Story," Cherish, . . .

80's GREATEST
Vol 24 (15 Pieces 50 Min)
Hill Street Blues Theme, Chariots Of Fire Theme, Dynasty Theme, . . .

70's GREATEST
Vol 12 (19 Pieces 45 Min)
Tie A Yellow Ribbon On The Old Oak Tree, We've Only Just Begun, . . .

60's GREATEST
Vol 13 (19 Pieces 45 Min)
Windy, By The Time I Get To Phoenix, Come Saturday Morning, . . .

GOLD & PLATINUM HITS
Vol 45 (19 Pieces 60 Min)
Thriller, 99 Luft Balloons, California Girls, . . .

KENNY RODGERS HITS
Vol 39 (12 Pieces 45 Min)
Lady, Ruby, She Believes In Me, The Gambler, . . .

BILLY JOEL GREATEST HITS
Vol 43 (17 Pieces 65 Min)
Piano Man, Say Goodbye To Hollywood, Only The Good Die Young, . . .

COUNTRY CLASSICS Part 2
Vol 42 (13 Pieces 50 Min)
Ode To Billy Joe, Me and Bobby McGee, Country Roads, . . .

TV THEMES
Vol 37 (21 Pieces 35 Min)
Hill Street Blues, St. Elsewhere Theme, Masterpiece Theater Theme, . . .

MOVIE THEMES
Vol 19 (16 Pieces 40 Min)
MASH Theme, The Rose, Can You Read My Mind (Superman), . . .

BROADWAY'S THEMES
Vol 47 (25 Pieces 65 Min)
The Last Supper, Dr. Doolittle, The Old Dope Peddler, . . .

CHURCH MUSIC
Vol 23 (26 Piece 50 Min)
Amazing Grace, What A Friend We Have In Jesus, . . .

BARBERSHOP
Vol 22 (22 Pieces 45 Min)
Hello Dolly, Put On a Happy Face, Hey Look Me Over, . . .

RICHARD RODGERS SONGBOOK
Vol 18 (19 Pieces 40 Min)
Climb Every Mountain, DO-RE-MI, The Sound Of Music, . . .

NOSTALGIA
Vol 17 (22 Pieces 45 Min)
Let Me Call You Sweetheart, Ain't Misbehavin', On The Goodship Lollipop, . . .

CHRISTMAS
Vol 36 (24 pieces 50 Min)
O Little Town Of Bethlehem, Let It Snow, March Of The Toys, . . .

POLKA PARTY
Vol 33 (17 Pieces 40 Min)
Happy Polka, Pizzacato Polka, Betty Polka, . . .

We accept CASH, CHECK, COD, VISA and MASTER CARD orders.
Shipping and handling US and Canada $3.00
Shipping and handling outside the US and Canada $5.00
COD charge $2.00
Illinois residents add 6¼% sales tax.

VISA MasterCard

Dealer Inquires Invited

Speech Systems
38W255 DEERPATH ROAD
BATAVIA, ILLINOIS 60510
(312) 879-6880

Deluxe Music Construction Set is a trademark of Electronic Arts. Sonix is a trademark of Aegis. Music Studio is a trademark of Activision.

We Focus on Video

In the growing video market one company leads the way in top-quality, low-cost, computer effects software. Aegis. You'll find our products working at every level in the video field. Professionals in television, video production, cable TV, industrial and government video, colleges, schools, home — you name it, we're there.

We're there because our products provide flexible technology that sparks the imagination. Technology that gets the job done.

You'll find our products being used for television and film at Prism Computer Graphics, for video publishing and exhibition systems at MetaVision, and for corporate video clients at both Master Communications and Pixelight. And that's only the beginning of the list.

We're there with products for graphics, animation, and sound

We handle metamorphic, cel, and color cycling animation with **Aegis Animator**. Animations are created and edited in an interactive environment allowing you to watch your animation as you build it. Winner of the 1986 CES Award of Excellence, Animator has already found thousands of uses around the world. Included in the $139.95 price is the Images paint system.

For the advanced animator **VideoScape 3D** provides an environment rich in 3-dimensional capabilities. Object motion and metamorphosis, camera motion, light sources, IFF foregrounds and backgrounds, and the ability to create animations in the ANIM format are just a few of the features that make the $199.95 price tag a great buy.

For titling you can't beat **Aegis VideoTitler**. It supports all of the Amiga fonts as well as its own polytext fonts, works in four different resolutions and uses overscan. It has 20 different styles, works with IFF, uses halfbrite if available, and supports the ANIM format. Included in the amazing $99.95 price is a slideshow generator that can mix ANIM animations with slides.

Our video effects don't stop with great visuals. Programs like **Sonix** take on the world of music and sound as well. Winner of a CES 1987 Award of Merit, Sonix lets you create your own instruments, compose music, and work with MIDI instruments. All for $79.95.

For more information and the dealer nearest you: (213) 392-9972

or to order direct: **1-800-345-9871**

AEGIS
DEVELOPMENT

2210 WILSHIRE BLVD. #277
SANTA MONICA, CA 90403

The Ultimate Video Accessory

Video and Computer Technologies have long been on a collision course. The Amiga may be just the tool to bridge the remaining gap.

by Larry White

Every time you turn on your TV, you see computer-generated graphics. These range from simple scrolling credits to spinning, twisting, shining, three-dimensional station logos, to the computerized Max Headroom himself. You've probably seen demonstrations of the Amiga's advanced graphics and animation capabilities, and wondered if you could use your computer to add sophisticated special effects to your own videos. You can!

Have you ever wanted to make your own cartoons? Maybe you've daydreamed about turning yourself into a superhero who flies off to save the universe.

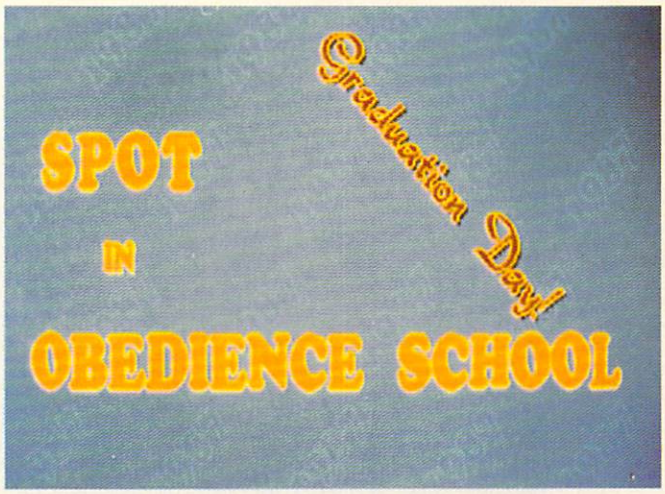
Perhaps you've been looking for a more effective way to use computer-generated pie charts and bar graphs in your next business presentation. Did you know that the Amiga can bring life to your presentations?

These and similar applications have spawned a new sub-industry: desktop video. Over the next few months, we'll examine desktop video. I'll try to answer most of the basic questions: What is desktop video? Why would anyone want to try it? What are the hardware and software requirements? Are there money-making opportunities? What is the role of the Amiga in desktop video? I'll take you on a step by step tour through the video making process, using fairly common computer hardware, software, and widely available consumer grade video equipment.

In this article we'll take a general look at desktop video. I'll explain the elements of video production, the various types of video and computer hardware, and some video theory.

Desktop Video: A Definition

For the purpose of these articles, I'll use the widest possible definition of desktop video. Video is defined as the representation of an image in an electronic form. While the original definition was specifically related to television (the transmission of still or moving images via radiowaves for viewing at a distance from the origination point), recent technological advances have broadened the use of the term to include most forms of electronic imaging that is ultimately displayed on a television or monitor. Thus the cuxrrent crop of video devices would include still electronic photographic cameras, and computers which electronically display text or graphic images.



Add a professional touch to home videos by adding title screens. This one was created using TV Text and Zuma fonts (Brown-Wagh).

While the term "video" technically refers only to the picture portion of the end result, many video applications would be far less effective or even useless without an accompanying sound (audio) track to provide narration, dialogue, sound effects, and music when appropriate. For our purposes, I'll include audio effects under the desktop video banner.

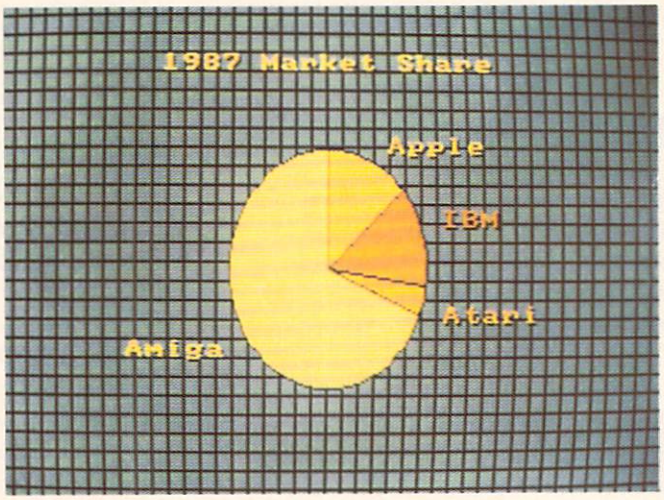
There are several other processes used in the preparation of a video, besides generating and recording the picture and sound. Depending on the desired length and complexity of the finished product, scripting, cataloguing images, and creating storyboards might all be part of the video creation process. Since the Amiga can be helpful in each of these areas, we'll include these under our "desktop video" banner also.

The final form of a desktop video is usually a video tape. Since almost any video signal can be recorded on tape (although some intermediate electronic conversion may be needed for compatibility), we'll also include presentations and animations produced directly on the monitor (since we can always play them back and tape them later).

(continued)

The last area of desktop video is production. Video production includes control of video cassette recorders (VCRs) for taping, playback, and editing. Additionally synchronizing audio effects and music are part of the production process.

In the past, coordinating the various components I've described might have taken a roomful of expensive video equipment, and often included a fully equipped, sound-proof recording studio and a large, expensive computer system (typically costing anywhere from $25,000 to $1,0000,000).



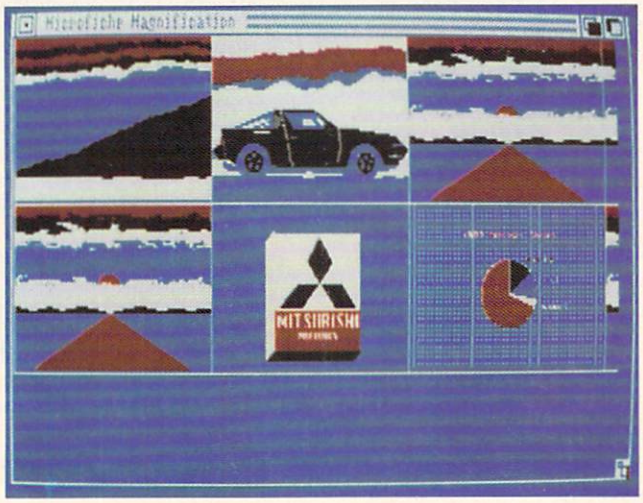
Add punch to a business presentation by animating the construction of your charts. Sound effects and background music can help convey your point. This sequence was taken from a Deluxe Video demo (Electronic Arts). If the final presentation can't be made directly on computer, transfer it to video tape.

Now many of these functions can be handled by a few thousand dollars' worth of video and computer equipment, especially when that computer is an Amiga. This is desktop video.

Applications

If I were describing desktop publishing, this section would be easy. Almost anyone can see the practical side of using a small machine for creating and printing newsletters, books, and brochures. While video plays a major role in all our modern lives, it's still somewhat difficult to recognize many of the practical uses for desktop video.

Many of you use computers to access bulletin boards. If you've ever flipped through the channels on a cable TV system, you've probably noticed something similar. Most cable stations have at least one channel which constantly displays local news, ads, and announcements. If you've watched television news programs, you've seen screens listing the daily sports scores or a five-day weather forecast. The simplest form of desktop video is a screen filled with type.



Storyboards help you plan better videos and presentations. Microfiche Filer (Software Visions) can display many IFF files at a time in up to 4 colors. A full image can be displayed within seconds. This sequence was captured from Deluxe Video demo (Electronic Arts) by Grabbit (Discovery Software).

The networks and larger stations may create these displays using expensive electronic "paint boxes," which are considered state-of-the-art. Smaller stations, however, have been turning toward lower-cost solutions, using a computer as simple as the Commodore 64 as a character generator. With a more sophisticated computer, such as the Amiga, character generator software can create scrolling and page cycling, using various fonts, colors, and shading. Some can even produce simple animations, providing a professional effect at an extremely low cost.

A character generator can also be used as a video titler. Give your home videos a professional touch by inserting a title segment which identifies the subject, time, and date of the video. Using a genlock accessory (which I'll describe in detail later), you can superimpose the characters directly over your existing live action.

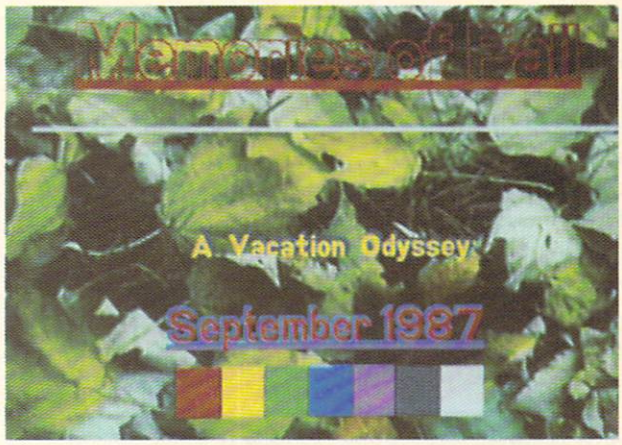
Desktop video can also become a major part of business presentations. Whether the presentation is displayed from a computer monitor, shown on a large screen by a video projector, or taped and replayed on a VCR, computer graphics can be mixed with sound effects, appropriate music, and narration to hold viewers' attention and emphasize your point. Either in-house departments or outside services can prepare a desktop video presentation. With a blending of titling, live video, and computer generated lists and graphs, can video corporate reports be far behind?

Education and entertainment applications also exist for desktop video. The vast base of installed VCRs creates a potential market for many video products. Many schools have already produced video yearbooks; these can be greatly enhanced by adding computer graphics generated by a desktop video setup. We'll get into many of these later, but let's take a look at the basic hardware and software requirements.

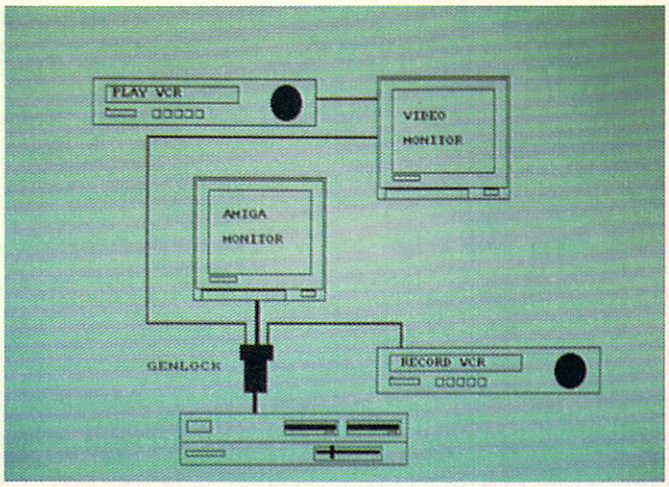
Hardware

Your basic hardware requirements will vary to some degree according to your specific application goals and software choices. I am currently using an Amiga 500 with 1 megabyte of RAM, an external 3 1/2 inch drive, and an Amiga 1080 monitor.

All three Amigas have National TV System Committee (NTSC) compatible output jacks; NTSC is the standard for television signal in the U.S. and several other countries. The Amiga 500 and 2000, however, output the luminance signal only, producing a black and white picture. You can record this signal, but you'll want to use a peripheral device to pull the luminance and chroma (color) signal from the analog RGB port which connects your computer to the monitor.



Mix computer graphics with live (or recorded) video for maximum impact. Genlock synchronizes video signals for a steady picture.



Record genlocked images on a second VCR. A second monitor is useful, but not essential. Some genlock devices (such as Mimetics's Imagen) won't genlock to an RGB monitor. To see a genlocked image on an Amiga Monitor, switch to composite video input.

At this writing, several such devices have been demonstrated and are expected to be available by the time you read this, but I still haven't received one for my own use. These include Creative Microsystems' V-1 Video interface ($54.95), and Mimetics Imagen (with genlock-ability to synchronize several video signals, $179). Several high-end units are expected shortly with even more special features, such as video fade control, but we'll get to these a little later.

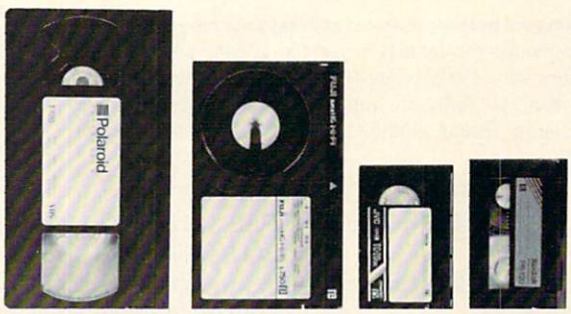
Additional RAM and a hard disk are desirable options, especially if you intend to do 3D animation, but you can accomplish most tasks with a basic system like mine. If desktop video is your only Amiga application, you may not need to purchase a printer unless you want to produce scripts for your live video portions.

Any format VCR with NTSC inputs can be used. If you intend to mix video with Amiga graphics, you'll need a second VCR or a camcorder (a video camera with a built-in recorder) that has playback capability (most do).

To produce professional desktop videos, you'll need a professional VCR with a flying erase head (also common on 8mm video systems), a single frame jog mode (usually controlled by a large knob on the front of the VCR), and compatibility with an edit controller. Such units are available from Panasonic (VHS) and Sony (BETA). If you're really serious, you may want to invest in the new Super VHS or soon to be released ED (extended definition) BETA units, which offer higher resolution, which is particularly valuable if you intend to distribute copies of your finished video.

If you use genlock, an extra monitor or TV can be useful for previewing the tape or live video signal. There are many other gadgets that you might want to add at this end of the system, such as signal enhancers, editor/switchers, and even additional VCRs. I'll discuss this equipment in another article.

(continued)



Any video format can be used for desktop video. Although physically incompatible, all formats are electrically compatible when taken as NTSC composite video from a VCR. VHS (left) is dominant in the market. Beta formats (second from left) offer a slightly better image. VHCS-C (second from right) is popular for cam-corder use and can be played in standard VHS machines with an adapter. 8mm Video (right) offers great potential and has excellent sound capabilities.

Software

There's quite a variety of software available for Amiga desktop video applications. In upcoming articles, I'll describe a variety of programs and their various desktop video applications. These will range from character generators to Deluxe Video (Electronic Arts) to 3D animations programs such as Videoscape 3D (Aegis), Animator Apprentice (Hash Enterprises), and the soon to be released Animate 3D (Byte by Byte).

•AC•

About the Author

Larry White is Technical Director of *Popular Photography* magazine. He has designed and constructed various test procedures and apparatus for testing photographic and video equipment. He is currently experimenting and investigating applications for desktop video. If you have a unique or interesting desktop video application, please write him via this magazine.

Software Suppliers:

Aegis Development, Inc.
2115 Pico Blvd.
Santa Monica, CA. 94043

Brown-Wagh Publishing
100 Verona Court
Los Gatos, CA. 95030

Byte by Byte
Aboretum Plaza II
9442 Capital of Texas Highway North
Suite 150
Austin, TX 78759

Digital Solutions Inc.
30 Wertheim Court, Unit 2
Richmond Hill, Ont. L4B189

Discovery Software International
903 East Willow Grove Ave.
Wyndmoor, PA. 1918

Electronic Arts
1820 Gateway Drive
San Mateo, CA. 94404

Hash Enterprises
14201 SE 16th Circle
Vancouver, WA. 98684

Software Visions, Inc.
26 Forest Rd.
Framingham, MA. 01701

Hardware Suppliers:

Creative Microsystems, Inc.
10110 SW Nimbus #B1
Portland, OR

JVC Company of America
41 Slater Drive
Elmwood Park, NJ 07407

Mimetics Corporation
P.O. Box 60238 Sta. A
Palo Alto, CA 94306

Panasonic Company
Matsushita Electric Corp of America
One Panasonic Way
Secaucus, NJ 07094

Sony Corporation of America
Sony Drive
Park Ridge, NJ 07656

The Sony Connection

Mate a Sony KV-20XBR 20-inch TV/monitor with your Amiga.

by Stewart Cobb

Some of us don't like small monitors. When I start programming, I kick back in my easy chair, put the keyboard on my lap, and stare down past my feet at the screen. Unless you have the eyes of a hawk, you can't see a thing from this vantage point with a standard twelve-inch monitor.

Perhaps you program the way I do. Or perhaps you need to teach a class or make a presentation with your Amiga. Perhaps you just want to show off the Amiga's stunning graphics on as large a screen as possible. In any case, if you're interested in a large, high-quality monitor for your Amiga, this article is for you.

What Kind of Monitor?

It was not too long ago that TV had no video inputs at all. Back then, a VCR or video game had to contain an "RF modulator" (a sort of miniature broadcasting station) in order to show up on your home TV. Most of these devices worked on channels 2, 3, or 4. They tended to produce fuzzy pictures because it takes much expensive equipment to produce a clean broadcast TV signal.

With the explosion of VCRs, videodisks, and other consumer video gear, the makers of high-end TVs started to offer direct video inputs. These inputs bypassed the TV tuner and fed directly into the video circuits. By connecting your video signal to this input, you could avoid two stages of signal processing (modulating and tuning) and in turn, get a better picture.

The video input wants to see a standard color TV signal. The standard color TV signal has a bandwidth of about 3 megahertz. A monochrome (black-and-white) TV signal can have a much higher bandwidth, but the manner in which the color is added to the signal drastically chops the bandwidth. The video inputs on an upscale TV are designed to accept this 3 Mhz color signal, but that's all they can handle. Unfortunately, that's just not good enough for a computer.

> In combination, the three "analog RGB" signals give the Amiga 4096 colors— a vast improvement over the PC.

Ordinary computer text requires a bandwidth of more than 20 megahertz. You need a true "video monitor" to display a computer signal. The first video monitors were monochrome because a monochrome TV signal has no bandwidth limits. Recently, we have started to send color pictures with a set of three parallel monochrome signals, one for each of the primary color (red, green, and blue). This process is called "RGB video."

The IBM PC and its clones use a "digital RGB" system. In such a system, each pixel in each video signal can only be on or off. "Digital RGB" gives the PC clones a total of eight possible colors (actually sixteen—they can also send a separate "intensity" channel in addition to red, green, and blue). The Amiga is more sophisticated: each pixel in each signal can be chosen from sixteen shades of grey. In combination, the three "analog RGB" signals give the Amiga 4096 colors—a vast improvement over the PC.

To use the full capabilities of the Amiga, we must pair it with an analog RGB monitor. The monitor sold with the Amiga is such a monitor, but otherwise, they are scarce. Nearly all color monitors sold for computers and TVs with "RGB monitor" capability are designed to work with digital RGB signals. Why? Because that's what IBM uses and IBM is the standard. But all is not lost ... Enter Sony!

Sony TVs contain the only Analog RGB monitors I have found in the consumer market. Some of them are small (9 or 12 inches) and are sold in computer stores. More interesting are the 20 and 25 inch models, generally sold in television stores. The Sony family's distinguishing trait is a 34-pin connector on the back or side panel.

This connector is the key. It contains inputs for analog RGB, digital RGB, left and right audio, and a few other goodies. Everything you need to hook up your Amiga is right there. All you have to do now is figure out how to use it. That's what this article is about.

(continued)

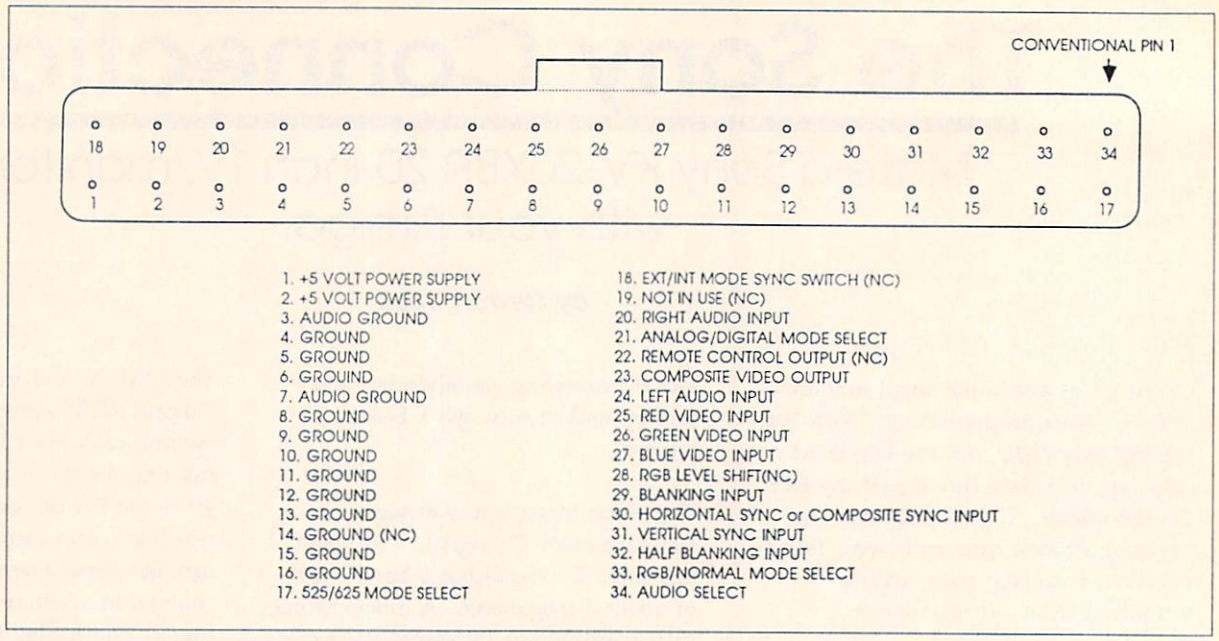


Figure One: *Pinout of the Sony video connector. The "hump" on the left side of the connector is the keying ridge. Pins labeled "(NC)" are not connected inside the KV-20XBR.*

The Magic Plug

I like large monitors. I came home last spring with a Sony KV-20XBR 20-inch TV/monitor, destined to become a mate for my Amiga. I thought I'd spend an hour or so hooking it up and then I'd start large-scale computing. Boy, was I wrong.

The problem wasn't the TV and it wasn't the Amiga. That 34-pin connector has everything you need to hook the two together. The problem is, they never tell you how to use it!

The manual shipped with the TV lists the pins on the connector, but doesn't say anything about voltage, polarity, or impedance. The manual mentions the "official" PX-34 plug—but the instruction sheet shipped with that plug just gives more of the same. Even the service manual is vague. Nowhere does it say exactly how one is supposed to use the connector.

The service manual does contain a schematic for the TV. With that schematic and a few hours of experimentation, I learned how to make the Sony connection. The remainder of this article gives detailed instructions for hooking an Amiga to a Sony monitor (plus a few hints on related matters).

What Signals?

In order to get the Amiga's picture to the Sony's screen, we must connect the red, green, and blue analog video signals and the "composite sync" signal from the Amiga outputs, to the corresponding Sony inputs. That's the minimum required to get a picture.

Standard analog video input and output ports are designed to present a 75-ohm impedance. If they are linked by 75-ohm coaxial cable, they are happy. If the cable or one of the ports does not show a 75-ohm impedance, the video signals will reflect and ring on the cable. You'll see this problem as a blurred, ghost-like picture. You don't have to measure these 75-ohm impedances, just follow the specs. (For those of you who want to know, the standard video signal level is one volt, peak-to-peak, in a 75-ohm system. The positive peaks are white and the negative peaks are sync pulses.)

The analog video outputs of the Amiga are designed to the 75-ohm specification. If you obtain 75-ohm coax, as I explain below, then the only thing left to consider is the Sony input.

The Amiga has separate pins for analog and digital RGB video outputs. The Sony gives you the option of analog or digital inputs, but it uses the same set of pins for both (pins 25, 26, and 27). You choose either analog or digital by setting the state of a separate switching pin.

Pin 21 on the Sony 34-pin connector is labeled "Analog/Digital Mode Select." When this pin is left unconnected, it puts the video and sync input pins into analog (75-ohm) input mode. This is exactly what we want to do to hook up an Amiga.

When pin 21 is connected to ground, the video and sync inputs go to digital mode. In digital mode, the pins react to TTL-level (+5 volts) inputs. The sync function must be provided by separate horizontal and vertical sync inputs. The digital mode matches an IBM PC-compatible color video output, with one exception: The Sony has no intensity input.

There are a few other switching pins on the connector whose functions may be of interest. Pin 33 is labeled "RGB/ Normal mode select." A monitor, which is also a TV (such as my KV-20XBR), uses Pin 33 as an override for the front-panel and remote controls. When this pin is left unconnected, the controls work as usual. When this pin is connected to +5 volts, it puts the Sony into RGB monitor mode, regardless of what the controls say. The effect is the same as pressing the RGB button on the front panel, except that now the process happens automatically. The Sony will remain in RGB mode as long as pin 33 is connected to +5 volts. I generally leave this pin unconnected because I prefer to use the remote control to switch modes.

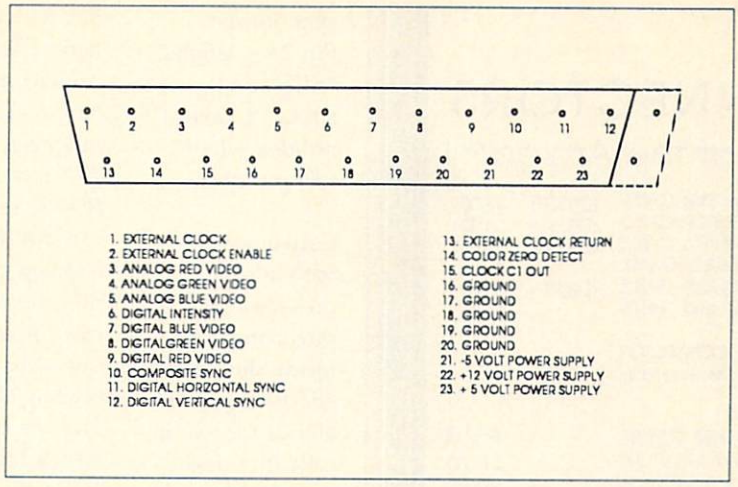


Figure Two:
The Amiga's video connector, as seen from the back of the Amiga. The dotted lines on the right show the piece to be cut off a DB-25 female connector to make it fit the Amiga's video port.

Pin 34 is labeled "Audio Select." This pin controls the external audio inputs on the 34-pin connector. When this pin is connected to +5 volts *and* the Sony is in RGB mode, the external left and right audio inputs (pins 24 and 20) are connected to the speakers. Otherwise, the audio inputs on the 34-pin connector are ignored and the speakers play the audio from whatever TV channel or video/audio input is selected (This is done "upstream" of the RGB switch.).

Again, I leave this pin unconnected so I can choose what to listen to (with the remote control) while I program. I've connected the Amiga's composite

(continued)

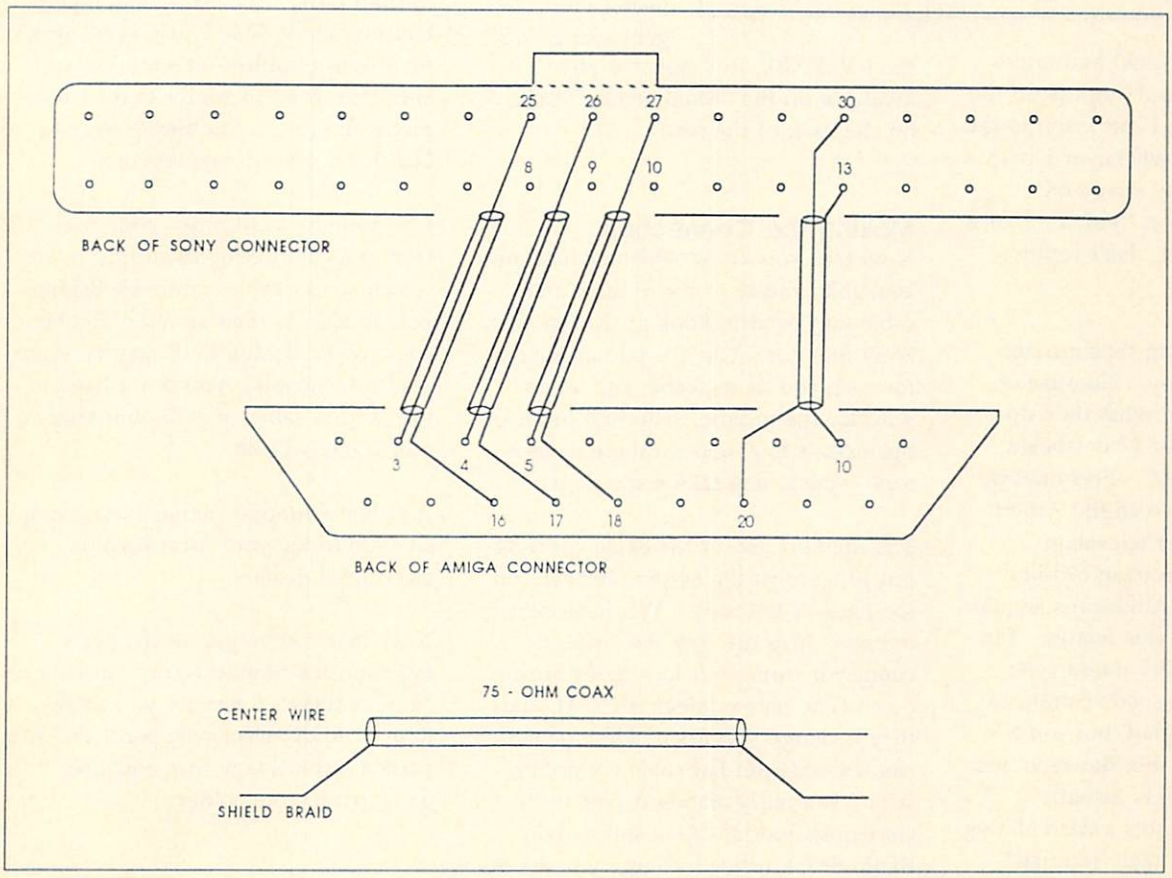


Figure Three:
The Amiga-to-Sony connection. The connectors are shown as seen from the back of the Amiga and the Sony. Make the indicated connections with 75-ohm coaxial cable.

video output (phono jack) and audio outputs to the "Video 3" inputs on the Sony. By doing this, I can listen to the Amiga's synthesizer whenever I wish. I can also listen to my stereo on "Video 2," my VCR on "Video 1," or a broadcast TV channel. Isn't technology wonderful?

A few of the switching pins are not connected on my Sony. Because of this, I cannot confirm what they do or how to use them. Pin 17 is labeled "525/625 mode select." Presumably, this pin switches between the American 525-line broadcast television standard and the European 625-line standard. Very few Americans should have to worry about this feature. Pin 18 is labeled "EXT/INT mode sync switch." The Amiga sends composite sync as a separate signal, but some analog RGB systems mix the sync into one of the video signals (usually green). Presumably, this switch allows the Sony to interpret that "internal" sync format. Pin 28 is labeled "RGB level shift." I have no idea what this pin does.

Conveniently, Sony also provides two pins connected inside the set to +5 volts (pins 1 and 2) for use in activating the switching pins. Also included is a "Composite Video Output" (pin 23) which indicates the video signal (TV or Video 1, 2, or 3) currently selected for output to the screen. This option could be useful for genlocking. On the KV-20XBR, this signal is also available on the "Monitor Out" jacks on the back of the set.

Making the Connection

Now that you know which signals are available, you're ready to build the cable you need to hook an Amiga to a Sony monitor. You'll need connectors for each end of the cable and some wire for the middle. The first order of business is to round up these connectors—which may take some doing.

The monitor end of the cable has a 34-pin plug, to fit the 34-pin "header" on the back of the Sony. You have two options: You can buy the "official" connector from your local Sony repair outfit (The part number is PX-34, and they'll charge you about $30.), or you can use a 34-pin flat cable connector, which is a fairly standard part in the electronics world. (One source is Radio Shack—they'll charge you about $3 for catalog number 276-1525.)

For the other end of the cable, you'll need a female 23-pin "subminiature D connector." This piece is extremely hard to find. Its 25-pin relative, however, sometimes called the "DB-25," is very common—it's the standard RS-232 serial interface connector. You can buy the DB-25 at Radio Shack (catalog number 276-1548). In a few minutes' work with a hacksaw, you can convert a 25-pin connector into one which will work just like the female 23. Remove pins 13 and 25 and the quarter-inch or so of plug which holds them. This may seem like a crude solution, but the only true 23-pin connectors I've seen in the Amiga world are the ones Commodore ships with its monitors.

Now that you have the connectors, you need the cable to join them. You need coaxial cable, commonly called "coax," with a characteristic impedance of 75 ohms. The standard 75-ohm coax is designated RG-59/U; that's what everyone uses for video work. If you walk into Radio Shack and ask for 75-ohm coax, that's what you'll get. Unfortunately, RG-59 just won't work for this application—it's too big and stiff. We need to solder four or five pieces of coax to one tiny connector. For that we need smaller coax.

One solution is to order your coax from a local electronics supply house which stocks cables made by Belden. Belden 9221 is their smallest 75-ohm coax, while Belden 8218 may be more readily available. You don't have to use Belden cable; any 75-ohm miniature coax will do.

Another solution—perhaps less expensive—is to try your local surplus electronics dealers.

Now that you've got all the parts together, it's time to figure out how to connect them. I assume you know how to solder electronic parts and you have a low-wattage iron and fine-gauge rosin-core solder.

If you decided to use a flat cable connector, you can make your job a lot easier through the following steps. Take the connector apart—there should be two or three plastic pieces. With a pair of long-nose pliers, extract all the little metal clips from the connector body. You may need to use some force. Don't worry if you destroy one or two clips; you won't need them all. Then, as you get ready to connect a wire, push the clip(s) for that wire into the appropriate positions on the connector. Insert the clips with the tabs along the *outside* of the connector and insert only the clips you're going to use. This adjustment will give you more room to work and will also reduce the possibility of shorts.

Take a good look at your 34-pin plug. You'll see a bump in the middle of one side of the plug. This bump is a keying ridge, which makes it near-impossible to insert the plug backwards. If you used a flat-cable plug, you'll also see a molded or painted marker (often a triangle) to the left of the keying ridge. This marker indicates pin 1 of the connector.

Unfortunately, Sony decided to be unusual. What everyone else calls pin 1, Sony calls pin 34. Even the way Sony counts up from pin 1 is different. If we were using flat cable for our connection, this would be a real problem. Since we're soldering our wires directly to the connectors, though, we can forget the numbers altogether and work from pictures.

Figure 1 should match the connector you see on the back of your Sony monitor. Notice the alignment notch and the 34 pins. I've numbered the pins in this diagram according to Sony's numbering scheme. I've also included a little triangle marker which points to the pin that everyone else calls pin 1. Note that most of the pins on the right side are connected to ground. The braided shield wires of your video coax cables should be soldered to these pins. Pins 3 and 7 are special "audio grounds" for the shields of the audio cables (if you use them). Keeping the video and audio grounds separate reduces the noise in your speakers.

Figure 2 shows the 23-pin Amiga video connector, as seen from the back of the Amiga. This figure is labeled with the Amiga pin connections. We want to connect to the analog video outputs on pins 3, 4, and 5 and the composite sync output on pin 10. You'll solder the coax shield braids to ground on pins 16 through 20.

Figure 3 shows a pictorial diagram of the connections we want to make. On the left is the Amiga connector; on the right, the Sony. The two are linked by the coaxial cables. You don't need to solder the coax shields to the particular ground pins I've shown—any grounds will do.

To prepare a piece of coax, cut the outer jacket back about a half inch. Be careful cutting through the outer insulator, so you don't nick the braid underneath. If you bend the cable sharply, the outer jacket stretches and is easier to cut. Peel the jacket off and brush the braid out straight. Next, twist the wire together to form one wire—this is the shield wire, which you will solder to ground. Cut through the inner insulator about a quarter inch back from the tip, again, without nicking the wire inside. Slide the bit of insulator off the center wire. This is the signal wire, which you'll solder to a video signal pin.

Measure four pieces of coax to the length you want. Remember, it's easier to shorten a wire than lengthen one. Prepare the ends of the coax as described above and solder them to the connectors as shown in Figure 3. When you've got those four wires done, plug the connectors into the Amiga and the Sony and make sure that the cable works. If not, check your connections against the figures in this article and the manuals which came with your equipment.

Once you've got the cable working, add any other wires you wish, such as sound or switching. Then fit the Amiga connector into a metallized shielded hood to reduce interference. Install the cable once more, check it out, and you're finished!

If you have any questions or comments, you can contact me care of this magazine.

•AC•

The Hack & Slay-Team presents:

Watch out
for
Snake Design!

We offer you:

* In every game a bonus point.
* Collect the points and earn:

for 3 points a Snake buttom.

for 10 points an arcade game for free.

See the games in our next advertisement.

Only software from Snake Design offers you this features!!!

A new Software Generation!

Witchcraft 49,95

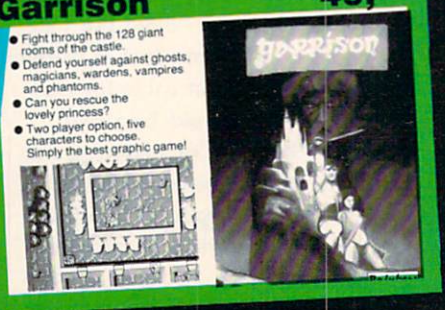
- Bored of destroying evil wizards? We got a more intelligent plot!
- Lots of mindbuggling logical puzzles, riddles and all those monsters lurking in the dungeon.
- More monster graphics than in any other fantasy game!

Emetic Skimmer 34,95

- Steer your spacecraft through endless tunnels and deep into the electronic caves of the supercomputers security system.
- Avoid hazards, destroy bases and finally find the deadly super-computer-complex.
- Featuring two disks full of graphic and action!

Garrison 49,95

- Fight through the 128 giant rooms of the castle.
- Defend yourself against ghosts, magicians, wardens, vampires and phantoms.
- Can you rescue the lovely princess?
- Two player option, five characters to choose. Simply the best graphic game!

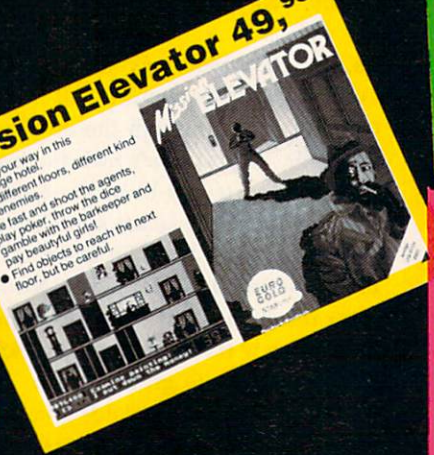


Spaceport 39,95

- Prove your skill steering the helicopter.
- Rescue the people in the mines.
- Destroy enemies and walls.
- Find the aliens power station.

Mission Elevator 49,95

- Find your way in this strange hotel.
- 64 different floors, different kind of enemies.
- Be fast and shoot the agents, play poker, throw the dice gamble with the barkeeper and pay beautyful girls!
- Find objects to reach the next floor, but be careful.



Detonator 39,95

- Zzap, catch and bang through 50 levels of pure fun.
- Digital speech, sound and music.
- Collect extra actions & weapons.
- Scrolling background graphics supports this ultimate game.
- Bonus levels in true 3D, the game in the game!

Available at these addresses:

C. Ltd
723 East Skinner
Wichita, Kansas 67211
Phone: 316 267 3807
Fax: 316 267 0111

American Software Distributors
R. R. 1, Box 290, Building 3
Urbana, Illinois 61801
Phone: 800-225-7941 or
217 643 2050
Fax: 217 643 2049

Impulse Inc.
6870 Shingle Creek Parkway 112
Minneapolis, Minnesota 55430
Phone: 800-328-0184 or
612 566 0221
Fax: 612 566 1822

Mindware International
110 Dunlop St. W. Box 22185
Barrie, Ontario
Canada, L4M5R3

Snake Design Software
Phone: 01149-69-7071102
Fax: 01149-69-708525

Calligrapher

from InterActive Softworks

by John Foust

If the Amiga Workbench comes with the FontEd program, why would anyone buy another font editor? Well, you just might want Calligrapher, a color font editor from InterActive Softworks, because it goes far beyond FontEd, and adds color, too.

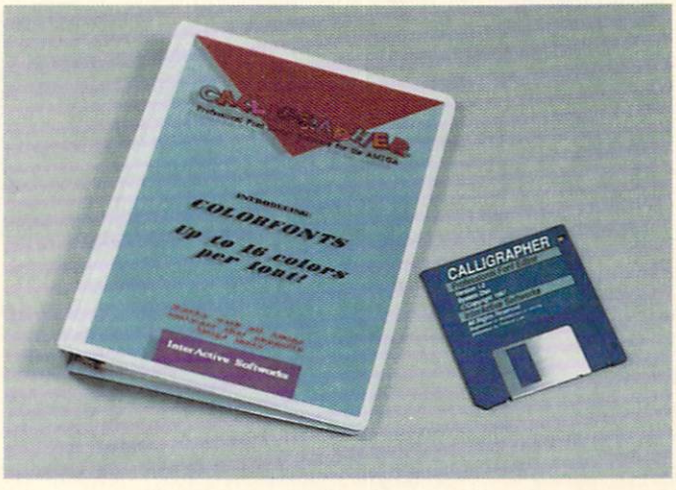
Standard Amiga fonts are made up of dots of only two colors. One color is essentially "no color," or clear pixels. The other color, for solid pixels, is determined by your Workbench colors or a color chosen from a palette, as done in Deluxe Paint.

The creators of Calligrapher extended the programming definition of Amiga fonts to include color (For more information, see *Amazing Computing* Volume 2, Number 8, "The ColorFont Standard."). The ColorFonts standard allows the dots that make up a character to be colored from a palette of up to sixteen colors.

Because of a wonderfully flexible programming feature of the Amiga operating system, you can use ColorFonts in existing programs, such as Deluxe Paint. By starting a patch program (included with your purchase of Calligrapher), any Amiga program that supports color and fonts can now access fonts created with Calligrapher.

InterActive has created a freely-distributable demo program, available in public domain disk collections and on bulletin board systems. The demo is crippled, though—it won't type vowels. All other characters work. Two demo disks, available from InterActive for ten dollars, demonstrate Calligrapher.

Calligrapher was first shown at the Commodore Show in San Francisco in February 1987 and was shipped later that summer.



User Interface

Calligrapher's user interface seemed imposing to me at first. I saw Jeff Braun of InterActive Softworks demonstrate it at several Amiga shows, and he made it seem very easy to use—then again, all programs look easy in the hands of their producer. As with most programs, an hour or two of experimentation made it much clearer to me.

The program is separated into different screens that work together: Font, Select, Edit, Effects, Style, and Pattern. You can jump between areas with function keys, as well as menu choices.

Font and Select Screens

The Font screen selects between the current fonts on disk and in memory. A simple editor window is provided for testing fonts in any screen resolution. Statistics about the current font are also shown in this window. In the Select screen, simple clicks and drags are used to select ranges of the present character set. This selection is used in other parts of the program.

Edit Screen

The Edit screen provides a mini-Deluxe Paint-type interface for modifying the pixels in font characters. IFF brushes can be loaded, stamped, and rotated within the space of a character. The commands are similar to Deluxe Paint, making Edit fairly easy to use. Along with Box and Circle tools, an arc tool unlike any other I've seen is included. You specify only one endpoint, then move to the other (instead of selecting two points), and then make the third in the middle. Unlike the Undo in Deluxe Paint, the Undo button here works only once. Once you've undone some-

(continued)

thing, you can't get it back by pressing it again. A Magnify gadget lets you zoom in on the character. The greater-than and less-than keys magnify the image when Magnify is selected.

In the Edit screen, the spacing between letters can be adjusted. Two inter-character spacings, the amount to backspace before drawing the character, and the amount to space after drawing the character, can all be adjusted. Normally, both amounts are near zero. Setting these values to anything else causes characters to overlap, making overstrike characters.

Effects Screen

FontEd includes controls for scrolling a character within the bounds of its shape. The Effects screen performs this same action on groups of characters. Kerning and spacing can be set for individual characters or groups of characters. The Effects screen also allows you to slant characters.

Style Screen

The Style screen adds layers and patterns to fonts. A layer is the outline of the font, in the same shape of each character. It can be a different color than the font. Outlines and shadows are forms of layers. If the layer is slightly larger than the current font, and placed below the current font at an offset, it forms a shadow. If the outline is directly below the current characters, it forms an outline.

Embossed effects can easily be created in this way (People have been doing this by hand in Deluxe Paint since the beginning, by entering text, clipping it as a brush, then stamping offset and enlarged versions of the brush.). The outline size and offset position settings define a style. Styles can be saved to disk and recalled. A style can have as many as sixteen layers. Layers can be merged, just like fonts or parts of fonts can be merged. By specifying the layering order of the merge, stunning effects can be created.

Pattern Screen

Patterns allow you to color layers automatically, like Fill patterns in Deluxe Paint. Patterns can be created from IFF brushes. For example, a brush that looks like bricks can be repeated to look like a wall of bricks. Calligrapher can stamp this pattern onto layers in several ways: in a regular fashion (as if they were cut out of a single bolt of cloth), in a random fashion, or from the center of the pattern.

Something called Transformations allows the changes made on a single character to affect the entire font. For example, you can resize the entire character set at once. On the Select screen, highlight all the characters in the set. Then access the Effects screen, resize one character, call up the transformation window with a function key, and press the Resize button. All the characters are now the new size. Some example transformations include the addition of new colors, layered shadows and outlines, italicization, and underlining.

As you can tell from this short description of the features of Calligrapher, making new versions of fonts is quick and easy. After learning the basics of one or two screens, I was able to load an existing Amiga font, color it, resize it, touch up the jaggy edges resulting from the resizing, and save it for use.

Large Fonts

Large type sizes, especially those with several colors, present several problems. A 37-point font of 96 characters in sixteen colors takes up 33,632 bytes on disk, and just as much in CHIP memory. Fonts must reside in CHIP memory, so a large font uses up much of the precious CHIP memory space needed by a paint program.

To guard against this space problem, the Calligrapher recommends creating a minimal number of characters in large sizes. It may be perfectly reasonable to create a large font with only five or ten characters. There is no need for any correlation between the letter on the keyboard and the shape of the character on the screen, so the letter A on the keyboard could make a capital R, if the font is drawn that way.

Font Problems

In my opinion, one of the serious deficiencies of the Amiga Workbench is the absence of icons for important files, such as fonts and printer drivers. Without icons for these files, or a utility for manipulating them, users are forced to the CLI. Try a "DIR FONTS: OPT A" sometime in the CLI, and see if you can guess how font files are arranged. The system of storing fonts on the Workbench is not documented for the novice user. Moving fonts and all their files is a complicated process which should be automated.

(continued on page 36)

Don't miss the boat...

with Amiga expansion products that limit expansion

A500 A1000 A2000

The Advantage™

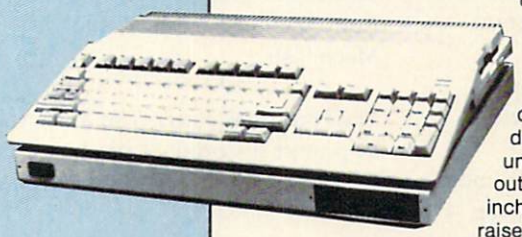
A two megabyte RAM expansion card.

1. No-wait-state design
2. Auto configures with Amiga software
3. Amiga standard design
4. User may install inexpensive 256x1 dynamic RAM
5. RAM disk software survives warm boot. (Not shareware or public domain. Included with purchase of memory board. Available separately for $19.95)

$199

A500 NEW! A1000

SubSystem™



Use cards designed for the A2000 with your A500, not out-of-date A1000 cards. The SubSystem gives you two expansion slots for A2000 cards and a space for an optional second floppy drive. The SubSystem fits under your 500, completely out of the way. Only 1.5 inches tall, the SubSystem raises the keyboard to the height of an average typewriter and actually makes it easier to use. A UL/CSA approved power supply is included that guarantees additional cards will not overtax the A500. The optional floppy drive is state-of-the-art CMOS design with extremely low power requirements. Cards and disk drive can be easily installed at a later date.

$249

A500 A1000 A2000

DIRECT MEMORY ACCESS **(DMA)** SCSI INTERFACE. Just because you have an Amiga 500 doesn't mean that you don't want the speed of DMA. Using the SubSystem with our A2000 card gives you what others only offer to A2000 owners. No matter what Amiga you own, Pacific Peripherals makes a SCSI for you. Our SCSI offers you compatibility with proven Apple Macintosh external storage devices. As a matter of fact, all of our drives are Macintosh compatible. If you use the SCSI in your A2000 you have an additional bonus... the ability to add a hard drive **inside** your Amiga and still use external devices. In addition to 30 megabyte and 50 megabyte (and larger) drives, Pacific Peripherals offers the Infinity removable media drive. Once you have purchased the Infinity, you have **unlimited capacity.** Each 10 megabytes of memory costs a whopping $18. (Does 100 megabytes for $180 sound more impressive?) With all this capacity you still get 75ms access time.

SCSI card	**$ 249**
SCSI card w/external drives:	
30 Megabyte	**$ 895**
50 Megabyte	**1295**
Infinity	**1295**

A1000

Cage II™

Two-slot expansion box for the Amiga 1000. Uses "zorro" standard cards **AND** passes thru the 86-pin bus for use with nonstandard applications. Power supply included.

$149

(415) 651-1905

Dealer Inquiries Welcome

Policy: Add 3% for VISA or Mastercard. Allow 3 weeks for checks to clear. Send cashiers checks or money orders for faster shipment. California residents add 7% for sales tax. No charge for UPS ground delivery. Next day and 2nd day delivery available. Prices subject to change.

Pacific Peripherals

P.O. Box 14575
Fremont, CA 94539

You can use the Calligrapher program to load a font from one disk and save it to another, but this may not be obvious to the novice. It took me a while to realize that this option was possible. Deleting fonts is another matter entirely, and must be done from the CLI. Deleting patterns and styles must be done by hand, too. With the comprehensive methods of creating fonts in Calligrapher, you may think they could have made a utility for manipulating and organizing sets of fonts. Pointing users to the CLI to move and delete fonts is not a pleasant alternative. Serious users of Calligrapher must understand the CLI, and I think that limits the program.

A program called FontFixer is provided with Calligrapher. Unfortunately, it is driven from the CLI instead of by icon. If you don't know how to do a DIR, you might never find it without an icon. FontFixer is fairly simple to use, if you understand command-line oriented programs ... but I don't think most users have this understanding. FontFixer checks the integrity of a given font directory. It does not move or copy fonts. It can remove unwanted font sizes and then remove empty font directories. The manual recommends using the CLI to manipulate font directories and files.

A utility called FontAssign is also available and is the equivalent of using the CLI "assign" command to redirect the system's notion of where font files are stored. On a freshly booted Workbench, the logical name FONTS: looks in the "fonts" directory on the system disk. With FontAssign, you can assign the FONTS: directory to somewhere else, such as a disk that contains your favorite fonts.

Another easy way of using other disks of fonts involves renaming these disks as "Fonts." This works with any Amiga program. If a disk named "Fonts" is in a drive, it will override the current FONTS: assignment (The same thing goes for disks named "C." If you insert a disk named "C," the system will look there for any command you type at the CLI prompt.).

Legal Questions

Calligrapher's ease of font creation introduces a legal question. What makes one font different from another? There are artists out there who create fonts for the Amiga and then sell disks of fonts. These fonts were created before the advent of easy-to-use font editors like Calligrapher. In some cases, they were created with custom font creation software or a font conversion program that ported fonts from another computer, such as the Macintosh.

The conflict is clear. The artist wants to protect his original product and investment in time and effort. Many users want to assemble large libraries of fonts. A font becomes "different" by changing the color or by adding a fill pattern. Both changes are trivial in Calligrapher.

> "...a necessity for anyone doing titling on the Amiga. FontEd is not enough."

According to one legal interpretation, the names of fonts can be copyrighted, but not the font itself. Many fonts may look like Helvetica, but have a different name. This same rationale is behind the gemstone names of the fonts on the Amiga. On the Macintosh, they used the names of major cities as font names, even though the fonts resembled popular commercial type faces such as Helvetica.

InterActive Softworks also sells disks of fonts. They contracted with several artists to make font sets for particular applications. One set is available for video titlers and graphic designers, another for newsletters, etc. The newsletter fonts are specially designed for use in bitmap font programs, such as PageSetter.

Several companies currently sell font disks. An early creator of color font disks was Lion Kuntz, who sells his "Lion's Amiga Art Studio" disks for $29.95 a pair. You get more than just disks of fonts. Kuntz has also written an excellent manual that outlines the best ways to use fonts in Amiga programs. I recommend these disks to anyone who wants to work with color fonts.

Summary

Calligrapher is a one-of-a-kind program. Because of its extensive features, I doubt it will see much competition in the future. Its $99.95 price tag will deter the casual weekend font designer, but a program like this is a necessity for anyone doing titling on the Amiga. FontEd is not enough.

•AC•

Calligrapher
List price: $99.95

Requires 512K,
at least 1 megabyte memory
recommended.

InterActive Softworks
57 Post Street, Suite 811
San Francisco, CA 94104
415) 956-2660

Lion's Fonts
$29.95 a pair

Lion Kuntz
PO Box 42252
San Francisco, CA 94142
(415) 431-1799

These Companies Joined
The Amiga Event in New York...

A–Squared
Activision, Inc.
Aegis Development
Amazing Computing
AmiNET
Amiga Sentry Magazine
AmigaWorld Magazine
AMuse, Inc.
Anakin Research
ASDG, Inc.
Associated Computer Services
Brown–Wagh Publishing
Byte by Byte
Commodore Computing Int'l.
Computer Living
Computer System Associates
Computer Living
Computer System Associates
Crystal Innovations
Designlab, Inc.
DigiTek, Inc.
Discovery Software
Electronic Arts
Enigma Publications
Firebird Licensees, Inc.
Fuller Computer Systems
Gold Disk Software, Inc.
Haitex Resources
Hugh's Software Ranch
Impulse, Inc.
Infinity Software, Inc.
Jumpdisk, Inc.
Lattice, Inc.
Magnetic Music, Inc.
Manx Software Systems
Meridian Software, Inc.
Microillusions
Micromagic, Inc.
MicroSearch, Inc.
Mimetics
Mission Graphics Support
NewTek, Inc.
New Horizons Software
New Wave Software
Oxxi, Inc.
PiM Publications, Inc
PS–Squared, Ltd.
Psygnosis Limited
Public Domain Software
R & DL Productions
ReadySoft, Inc.
Sedona Software
Software Insights
Software Visions
Sound Quest, Inc.
Spencer Organization
Spirit Technologies
subLogic Corporation
SunRize Industries
Supra Corporation
TeleGames
The Other Guys
Vertex Associates
Very Vivid, Ltd.
WordPerfect Corporation

Here's Your Opportunity to Join Us in Los Angeles!
January 16–18, 1987 at The Westin Bonaventure

YES! Pre–register me for AmiEXPO–L.A.
for ☐ Saturday ☐ Sunday ☐ Monday

Pre–register by **January 2, 1988** and save $5 off the on–site fee. Admission to seminars is $5 per seminar or 6 seminars for $25.
Please register me for __ 1 Day, $15
__ 2 Days, $20
__ 3 Days, $25

Please include __ seminar tickets @ $5 each
– or –
__ 6 Seminar tickets for $25

Name _____
Company _____
Address _____
_____
City _____ State ____ Zip _____
Make check or money order (U.S. funds only) payable to AmiEXPO and return to:

AmiEXPO
211 East 43rd Street, Suite 301
New York, NY 10017

NEW for the AMIGA

Precisely™

The Word Processor that doesn't try to be a desktop publisher.

Programs designed for graphic artists are difficult to learn, hard to use, tie up lots of memory, cost too much, and do not handle text very well. Graphics printing on a dot matrix or daisy wheel printer is very slow.

Introducing PRECISELY. The word processor optimized for people who work with words not pictures! Everything you expect in a word processor, such as:

- What you see is what you get (WYSIWYG)
- Very fast screen update/printer speed
- Easy to learn and simple to use
- Converts PaperClip, Pocket Writer and SpeedScript documents
- Online HELP • Mail Merge • Keyboard Macros
- Oops Key to undo mistakes
- Built in spelling checker coming soon

This is PRECISELY the word processor you've been looking for at PRECISELY the right price! Only $79.95 (sometimes less is more). Add $3.00 for shipping and handling, CA residents add 6% sales tax.

CONVERSION SERVICE

Do you have just a few C64/C128 disks of text or data to transfer and no 5.25-inch disk drive on your new Amiga? Use our new Disk Conversion Service to transfer the entire 1541/1571 disk image to a 3.5-inch Amiga disk. Then use Disk-2-Disk, with all of its translation features, to extract and convert individual Commodore files from the 1541/1571 disk image into standard Amiga file format.

The 1541/1571 Disk Conversion Service fee is $6.50 per disk plus a $15* service charge (per order). The fee includes a 3.5-inch diskette, and return shipping via UPS surface.

CAUTION – This conversion service can only process 1541/1571 disks which are formatted in the standard 35-track 256-byte sector format. Disks which are copy-protected, marked with a copyright notice, or formatted with non-standard formats cannot be converted to Amiga format.

* $15 service fee is waived when you purchase Disk-2-Disk with your order.

TRANSFER FILES

TRANSFER C64/C128 files to and from your Amiga!

DISK-2-DISK reads your PaperClip, SpeedScript and Pocket Writer documents or other files on floppy disk directly into your Amiga. Transfers all file types. Use these transferred files with your favorite Amiga programs.

- Reads/writes 1541/4040 and 1570/1571 disk formats.
- Converts Commodore/PET ASCII to Amiga ASCII and vice versa.

TRANSFER MS-DOS and ATARI ST files to and from your Amiga!

DOS-2-DOS reads Lotus 123 worksheets, wordprocessing documents or any other files on floppy disk directly into your Amiga for use with your favorite Amiga programs.

- Reads/writes both 5.25" AND 3.5" MS-DOS disks.
- Reads/writes 3.5" Atari ST diskettes (GEM format).
- Converts ASCII file line ending characters.

Disk-2-Disk requires the Amiga model 1020 5.25" disk drive. Dos-2-Dos runs on any standard Amiga. Disk-2-Disk $49.95, Dos-2-Dos $55.00. Add $3.00 for shipping and handling, CA residents add 6% sales tax.

CCS

Central Coast Software™

286 Bowie Drive, Los Osos, California 93402 • (805) 528-4906

MasterCard VISA

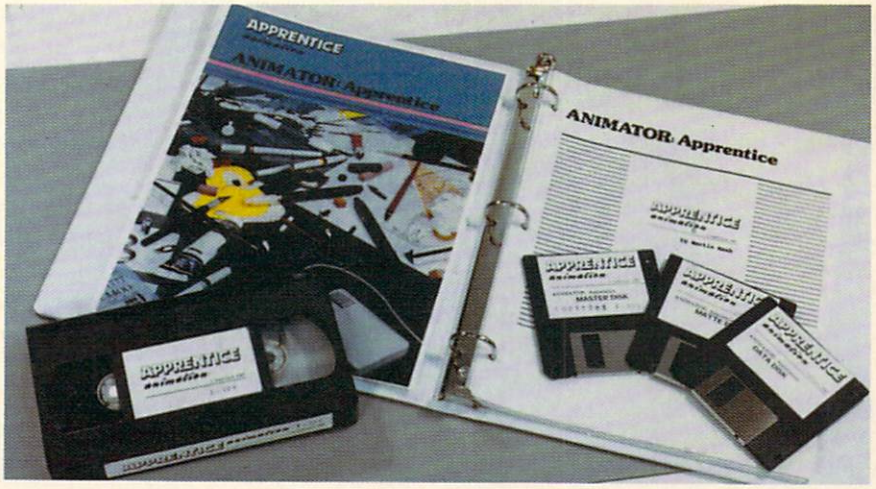
Animator: Apprentice

A full-fledged cartoon-style animation system

by John Foust

Animator: Apprentice from Hash Enterprises is a full-fledged cartoon-style animation system. In the Animator: Apprentice advertisements, the animation style is described as "Disney-like." In some sense, this is true.

Other animation programs, such as VideoScape 3D and Forms in Flight, use polygon-based objects; the animated objects look as if they were machined from steel because they are smooth and have sharp edges.



Animator: Apprentice characters have a decidedly organic look because the program uses a method of creating objects very different than polygons. An object is built of slices, like a loaf of bread, and the crusts take on color. You define the outline shape of each slice of bread, color its edges, stack the slices, and create part of an object.

The program is split into several modules: Character, Action, Director, Rehearsal, Record, Display, and Sculpt.

Character

The body of an Animator: Apprentice character is developed with the interaction between the Character and Sculpt modules. The Character module defines an armature. An armature is a bendable model of a skeleton often used by animators and artists. (The Animator: Apprentice manual doesn't use this term; it calls them bones.) With the Sculpt module, you design colored padding for the armature, defining the shape of the character. More on the Sculpt module later.

With the Character module, you define the interdependencies of the bones in the skeleton of the creature you want to create. The example given in the manual is good. If you are designing a human-like creature, the pelvis bone is the most primary bone. Move it, and the other bones follow. The torso and thighs are connected to the pelvis. If you move the leg, then the lower leg and foot move with it.

Bones are defined in tree fashion this way. You see a flowchart-like depiction of the bone family tree as well as the skeleton itself, drawn with straight lines. Each bone is given a possible range of motion in this module, too.

Action

The possible actions of an armature are defined with the Action module. You can select a bone and define a motion for it over time. By setting up motions for all the bones in a character, a larger sense of motion is defined.

The program comes with an example motion called "Walking." It uses a human biped armature. Even in the preview mode of the Action module, it is clear that Actions can be very life-like. In "Walking," the feet move up and down to touch the ground, the whole character moves up and down slightly as it walks, and the arms swing back and forth in a very natural way.

You can grab bones with the mouse and move them into position. With a motion sequencer screen, you can adjust the time and range of motion described for each bone. The program will also calculate "ease" for you, which is a way of making animated motions look more natural.

(continued)

Motion is hierarchical, too. If you move the upper arm of a character, the hands and fingers move along with it. Swivels and turns of the arm produce similar rotations in the subparts of the body.

After learning to use the Action module, I could respect the amount of effort expended to create the "Walking" motion. One beauty of this program is that pre-defined motions can be used with any character whose bone structure resembles the bones in the action. The motion is not particular to one character; it can be used on many characters.

The system of generalized armatures and motions is very powerful. This class-like structure is a generation ahead of every other Amiga animation program. Other programs rely on very specific motions and object descriptions that only apply to one object.

This flexibility demands a lot of detail from the user, too. A lot of thought and planning must go into each character and motion.

Director

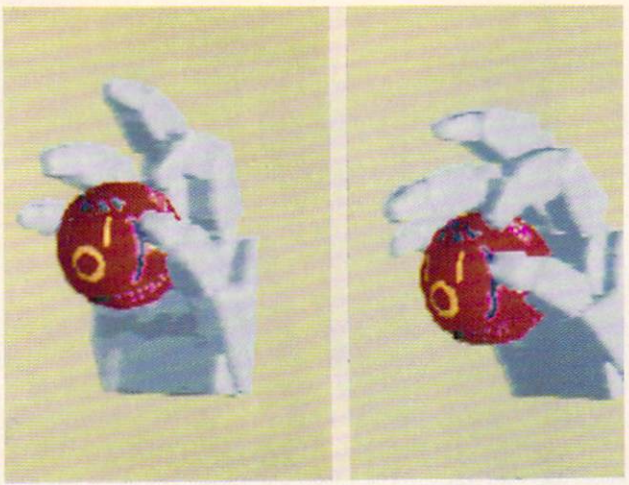
After you have defined armatures and motions, you can place them on a set with the Director. First you teach a character to walk, then you tell it where to walk using Director. You place the character on a stage and point it in a direction, and decide where and when it should be in a certain position. When it reaches that position, you can adjust its orientation (make it look another direction) or tilt it. Camera motions are available right down to trucks. Move-in and pull-out are also possible with Director.

Rehearsal

The Rehearsal module plays back a scene in a wire-frame mode. You can see characters moving on and off the stage with respect to the current camera viewpoint. The Director module can do this to a certain extent, too, so you don't have to switch between modules to see your scene.

Sculpt

After the armatures are created, the motions defined, and the motions linked together in a scene, you draw the body parts for the armatures with the Sculpt module. These body parts are called Segments. A Character has Segments that fit onto the bones created in the Character module. Segments might include forearms, thighs, a head.



Each Segment is created separately and made up of slices. I found it convenient to think of the slices as being made up of little blocks. The blocks are each given a color. Each slice of a Segment must be defined separately. This can be done in an intuitive and automatic way, but it soon bogs down in a lot of detail for complex characters.

Segments can be created semi-automatically by drawing IFF pictures that present views of the Segment, using a feature called AutoSculpt. The shape of the Segment is defined by two pictures called Visage and Profile that give the front and side outlines of the segment. If the Segment was a head, then Profile is a silhouette of the side of the face, showing the nose and chin, for instance, while the Visage shows a front view including the outlines of the ears.

The AutoSculpt feature interpolates a round or rectangular shape from the front and side views. This shell is then colored by hand or semi-automatically.

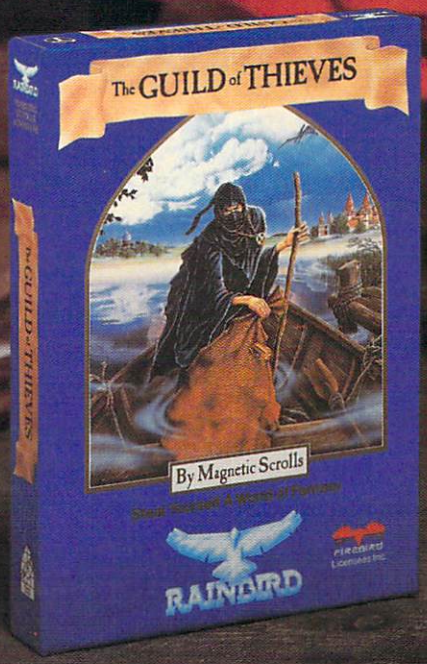
To color the shell automatically, you draw IFF pictures of the front and back of the Segment in a paint program. If this was for a head, you paint eyes, facial features, and hair on the previously used visage outline. The Texture feature superimposes the Front and Back mattes on the new Segment shape. You are free to touch up the shell colors with a built-in editor in the Sculpt module. This technique is very powerful, considering that the IFF pictures could be digitized images from Digi-View, touched up in Deluxe Paint. This technique produces very realistic looking characters.

The profile pictures must be drawn very carefully in a paint program. AutoSculpt is smart, but it can't create perfect shells. If the character is complex, you must do a lot of editing of the individual slices.

The user interface in the Sculpt module leaves a lot to be desired. I worked with the Sculpt module for several hours. A lot of the time, I could not figure out what was going on. It took a lot of mental energy to understand the manual. I shuffled between the videotape, the manual, and the program until I got it all right, yet I thought I still hadn't mastered the Sculpt module.

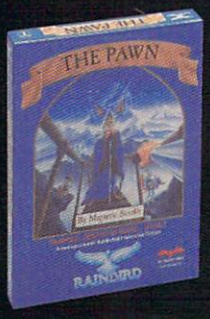
Currently, there is a limit of 32 colors per Segment. The final animation can

(continued on page 42)



This is the difference between reading an adventure and living one.

Picture the most exciting text-only Adventure in your software collection— WITH PICTURES!



GUILD OF THIEVES is the long-awaited sequel to THE PAWN, which came sizzling onto the software scene to awards and accolades. Both are available now from Firebird.

These stunning Adventures, completely different from any game you've ever played, seduce the sight with their breathtaking illustrations and engage the intellect with a truly revolutionary text-handling system. The most sophisticated parser on the market lets you input complicated sentences and interact with a whole cast of fascinating characters in the mythical kingdom of Kerovnia.

GUILD OF THIEVES and PAWN, Computer Entertainment Adventure of the Year, come to you from Firebird, publishers of world-famous leading edge games under the Rainbird label.

You'll find the Firebird logo on other addictive Adventures too, as well as on absorbing Strategies, realistic Simulations, and fast-paced action Arcade games.

Firebird brings the best in entertainment software to those who enjoy a whole range of interactive excitement. We'll prove to you that you don't have to keep switching brands to satisfy your obsession for challenge!

* (520ST Graphics)

FIREBIRD

"The First Full Line In Software"

Firebird Licensees, Inc.
P.O. Box 49, Ramsey, NJ 07446
(201) 444-5700

Available for C-64, Amiga, Atari 520ST, Atari 800/130, MacIntosh, IBM and compatibles. A "text-only" version is available for Apple II computers at $39.95.

be rendered in HAM 4096 color mode so characters with different color schemes can exist on the stage at once.

Assign Problems

I set up the Animator: Apprentice program on my hard disk. I set up ASSIGNs to point to the data directories needed by the program. Currently, the program uses disks with names such as "matte," "data," and "frames" to store things, but I set up logical names using ASSIGNs to other directories instead. I discovered that the Sculpt module had specific references to "DF1:" instead of following its own logical names. I noticed this when I tried to save my Segment. This means the program would fail if I put the "data" disk in the internal drive.

The same is true of the Record module. It accesses "DF0:" regardless of the current ASSIGNs. I used the public domain "strings" program to determine that the program did indeed reference the drive explicitly. I had a preliminary master of an AMICUS disk in the internal drive when the Record module wrote to the disk. I got very nervous when the drive light went on. I hot-patched the executables to change the "DF0:" to be something like "FOO:," then used ASSIGN to point "FOO:" at the correct directory.

Recording

After the animation is defined, the Record module makes the frames of the animation using the predefined motions and characters. It has two modes, shaded and unshaded. You can also select the number of colors used, including HAM. The unshaded mode makes animations that look like Saturday morning cartoons, with solid-filled two-dimensional looking shapes. Each Segment gets a black border.

The shaded mode makes more realistic characters, creating the "Disney-like" look, with no lines between Segments.

The frames can be saved to disk in a packed format for fast playback or in the IFF format for use in other programs. A view mode is another choice. With it, you can watch each frame of the animation as it is created, but it is not saved to disk. Depending on the complexity of the scene, it takes from several seconds to several minutes to create each frame of the animation.

I have noticed some glitches in HAM animations. Horizontal bands of color crop up in spaces on the character. This is distracting in a final animation. (This problem is endemic to the Amiga HAM mode in any program. I suspect all HAM animation programs will have trouble with this.)

The Record program uses a custom video mode that prevents you from accessing the Workbench while the frames of the animation are being produced. Amiga M and N did not bring back my Workbench screen. Your Amiga is effectively tied up while rendering frames. Also, programs such as Grabbit can't find the Record or the Display screens. It is an overscan display, and perhaps this is the reason.

Playback

The Display module plays back animations. It can single-step the animation or play it at different speeds. Hash Enterprises has made this program freely distributable so you can give away copies of your animations to people who do not own Animator: Apprentice.

Manual

In a previous AMICUS Network column, I said the Animator: Apprentice manual was "refreshing," in regard to Hash's perspectives on Amiga animation that accompany the manual.

(continued on page 44)

PACKED WITH POWER!

Three of the BEST utilities for your *AMIGA*® !

ZING!®

$79.95

ZING!® is a collection of ***AMIGA***® utilities which combine the powerful CLI commands with the friendly Workbench environment. Files can be displayed and manipulated (e.g. copying, moving, deleting) with the mouse. All of the basic system commands (available in CLI) have been carefully redesigned into mouse, menu, and function key operations. You can selectively copy files and directories from entire disks in a single step! In addition to enhancing and simplifying the normal capabilities, **ZING!** provides an integrated collection of new tools for the *AMIGA*. Included are a task monitor, print spooler, screen dump tools, and much more! Everyone, from the beginner to expert, will find using the *AMIGA* easier and more productive with **ZING!**

ZING!Keys® is a sophisticated reprogrammable MACRO and Hot Key program. A program which can be used with any programs in the multitasking environment. You can train **ZING!Keys** to accomplish the most annoyingly repetitive tasks in a much easier fashion. You can program any key stroke to type out any series of commands or text. You can even record mouse movements and play them back as a single key stroke! All MACROs and Hot Keys can be used from within any multi-tasking program on the *AMIGA*. Save time NOW by ordering **ZING!Keys**!

ZING! KEYS®

$49.95

THE DEMONSTRATOR

How many times have you demonstrated your *AMIGA* to your friends or potential buyers? **The Demonstrator** is the answer to your dreams! This new product will record your demonstrations and allow you to play them back automatically! You can add text windows, subsections, and speech to produce sophisticated tutorials. You can control the speed or STOP the playback, or lock out the keyboard so no one can interfere with the demonstration. You can cause the demonstration to repeat itself automatically! Buy **The Demonstrator** today! **Only $39.95!**

MERIDIAN SOFTWARE INC.
P.O. Box 890408
Houston, TX. 77289-0408

(713) 488-2144
Credit Cards and Dealer Inquiries Welcome!

Unfortunately, my comment was premature. While the extra essays that came with my review copy of the program were thought-provoking, the rest of the manual is woefully inadequate for learning the program. Its error is common in computer manuals: It describes the options available in the program, but does not give enough examples to illustrate those options clearly.

The manual has been through several revisions, and the update (in the form of a complete manual replacement) was sent to all owners. The changes to the manual involve added paragraphs and additions to describe new options. Unfortunately, no further examples were added. The manual remains very discontinuous.

Support

The price of Animator: Apprentice is high ($299), but for good reason: The support level is high. The author of the program is available for telephone support. Also, this past summer, Hash sent a two-hour videotape tutorial to all owners. This tape is now part of the product. So, along with the manual, you get a tape of the author himself describing the program and walking through the creation of a simple character. This was a great boon to my learning curve. It did take about six to eight hours for me to get up to speed with the program. Some of those hours were before the videotape arrived, and things went much faster after that.

Memory Requirements

Animator: Apprentice requires 512K of CHIP memory plus one megabyte of FAST expansion memory. It also eats disk space. Animations take a lot of space; a disk holds about ten to thirty seconds of animation, depending on the complexity of the scene. Segments can get large, too. I made a head that took up 160K on disk. Most were much smaller, though, about five hundred bytes to 15K.

Summary

I've studied all the Amiga animation programs, and I believe Animator: Apprentice is the most advanced in terms of object creation, motion control, and direction. I have seen some tremendous animations from Animator: Apprentice. Its style of animation is unlike any other.

This program is not for novices. It takes a lot of time to understand, and even more time to master. The price is high: You must invest both time and money to use this program well.

•AC•

Animator: Apprentice $299

Hash Enterprises
14201 S.E. 16th Circle
Vancouver, Wash 98684
(206) 256-8567

by John Steiner

Bug Bytes

The Bugs & Upgrades Column

Early Amiga 2000 units have a video problem. Only about 800 units are affected, and Commodore has quickly provided authorized Amiga service centers with the correction. The video problem involves a lack of clarity and resolution on the display. The defective video display is visibly inferior when compared to a properly working 2000.

If you suspect that your Amiga 2000 is defective, write down your unit serial number and call an authorized Amiga service agency. Each agency has a list of serial numbers of units that may require modification. The correction is a minor modification to the video output circuitry that should take a dealer only a few minutes to complete. Commodore is paying the service agency to make these repairs, which are completely under warranty to the end-user.

Last month, I reported on the upgrade to PageSetter from Gold Disk. I did not find out until after the column went to press that, unlike the original program, the upgraded PageSetter uses "look up a password in the manual" copy-protection. To access the new version's features, you must keep your manual close by. A couple of users in my area had trouble starting the program because the password is case sensitive. If the word is capitalized in the manual, you must capitalize it when you type it in. "Caps lock" must also be off when you enter the password.

I contacted Gold Disk about the newly-added copy-protection. Arno Krautter, one of the program developers, gave me the following reply (edited to save space):

"... We really had very little choice in the matter. The reality of the Amiga market is that there are more users in Europe than in the US. Almost all of our European distributors refused to sell the program without a form of copy-protection. The "look up the word in the manual" scheme seemed to us to be the least obnoxious scheme that was feasible. It allows a hard disk user to use it, [and it] allows you to make as many backups as you need, but unless someone goes to the trouble to copy the entire manual, [it] is useless to someone who copies it without buying it.

"...Piracy in Europe is so rampant that our distributors demanded something. We do not intend [copy-protection] to be a general policy and Professional Page will definitely not be protected as any professional product should be. I hope this at least lets you see the situation from our point of view, even if it doesn't change the situation much."

Micro-Systems Software's Online! 2.00 terminal program has a problem in file transfer mode. XMODEM transfers will not work if you set the AUTO-CHOP option of the FILE menu to OFF; it must be set ON. Micro-Systems Software has a fix available if you send your master disk. They are currently shipping version 2.01. If you bought Online! version 1 before 1987, the upgrade to 2.01 is $19.95; if you bought it after January 1, 1987, the upgrade is $9.95. Verification of purchase date with a copy of a purchase invoice or cancelled check is required.

Micro-Systems Software is also currently shipping: Scribble! version 2.01 and Analyze! version 2.03. Current users of version 2.0 of either of these packages can upgrade for free by returning the master disk. If you are using version 1.X of Scribble!, an upgrade from 1.0 to 2.01 is free; again, just send the master disk.

The upgrade from Analyze 1.X to 2.03 is $49.95. Version 2.X of Analyze! contains powerful Macro and Graphics functions not available in version 1.

All upgrades listed above also have a shipping and handling charge of $3.50. If you have any questions, call or write to the address below:

Micro-Systems Software
Technical Support
1279 W. Forest Hill Blvd
Suite 202
West Palm Beach, FL 33414
(305) 790-0772

Byte by Byte has announced that their 3D ray-tracing program, Sculpt 3D, has been upgraded to version 1.101. The new version has a new algorithm which speeds up ray-tracing by at least 3 times, and even faster in the GLASS and MIRROR object modes.

(continued)

As of this writing, registered owners who send their warranty cards automatically receive an upgrade disk from Byte by Byte. Users are asked to send $5 or fill out a survey form and return it with $3 to cover the postage cost of the upgrade. If you did not receive the upgrade, return your registration card and you will receive notification of all future upgrades. Byte by Byte is working on even faster algorithms and will be updating the software regularly.

For those of you with 68020/68881 combination boards, such as CSA's Turbo Amiga, a specially optimized upgrade is available that uses those coprocessors for maximum speed and efficiency. To order that version, send your registration number and a check for $30.00 to the address below:

Byte by Byte Software
Arboretum Plaza II
9442 Capital of Texas Highway North
Suite 150
Austin, TX 78759
(512) 343-4357

Textcraft Plus is finally available and original Textcraft owners can now upgrade. Textcraft Plus has been significantly improved over Textcraft. Major improvements include: the new version works under Kickstart V1.2; it supports multi-tasking (It runs in a window that can be sized, rather than a screen like the original.); it can access multiple drives, directories, and hard drives; and it supports international keymapping. All registered owners of Textcraft will be sent a postcard noting the upgrade policy, which is as follows:

To upgrade to Textcraft Plus, send your original Textcraft diskette and $35 to:

Commodore Promotion
Textcraft Plus
P.O. BOX 695
Holmes, PA 19043

You will be sent a new Textcraft Plus package.

Readers have turned up a few more bugs in Word Perfect. These bugs occur in the updated version of Word Perfect, released August 10, 1987. All are minor and can be easily avoided until a bug fix becomes available.

WordPerfect Corporation has confirmed that the double word feature of the spelling checker fails to find double words across hard returns. Double words are properly found across soft returns.

The print program of Word Perfect should delete the WP#? files on T:. T: is usually written to RAM disk, and these temporary files disappear upon power down. If you have rearranged T: to appear on a floppy or hard disk, WP#? files will not be properly deleted. WP Corporation is working on this bug.

Jacques Chatenay, a reader from Fargo, ND, has reported that there seems to be a bug in the Word Perfect Okimate 20 print driver. It seems to ignore the first set of tabs in a tabbed document during the first few print jobs. Printing the same file with another printer, using another printer driver, seems to correct the problem. If you have had similar problems, let me know, and we will see if we can document the problem more thoroughly.

There seems to be a problem with "Stop Current Job" in the Print Control menu. Choosing "Stop Current Job" locks up the printer. Use "Cancel Job" instead. Word Perfect Corporation is aware of this problem as well.

That's all for this month.

•AC•

Playing *DYNAMIC DRUMS* on the Amiga™

by David N. Blank

BLANK@BRANDEIS.bitnet
OGION@BRANDEIS.csnet

When I first saw New Wave Software's advertisement for Dynamic Drums, I had serious doubts that this new product could offer even a fraction of the performance of the $700 drum machine I was planning to buy. After working with the program for many enjoyable hours, I must say that it has far exceeded my highest hopes. The program has capabilities that most professional drum machines lack, for less than ten percent of the price of these other machines.

Let's begin our tour of this package with a look at its contents. Dynamic Drums includes a two disk set (disk one contains the program; disk two contains the drum samples), a short, but informative manual, and an introductory instruction cassette tape. The cassette tape is a nice touch you would expect to find only in the stellar priced business software packages. The tape provides a good, hands-on tutorial concerning the composing process.

Now, it is time to turn your favorite computer chair into a drum throne (Yes, that is the real name for a drummer's seat. If you don't believe me, ask another drummer.). A double click on the snare drum icon boots the program. The main screen fades in like a good movie, and after a short, but dramatic pause, a requester appears. The requester asks you to select the "drumkit" you would like to play. In Dynamic Drums, the term drumkit refers to a user-definable set of ten drum samples. Each sample is assigned to digits 0 through 9 on the numeric keypad. The custom keypad configuration can be saved for later use.

Dynamic Drums comes with approximately ten pre-defined drumkits, tailored to different music styles. Drumkits can also be created to suite different tastes. The program then loads the set of samples and displays the keypad bindings in a window. Once this work is done, playing the drum samples in real-time is as easy as pressing keys on the keypad! Since the Amiga has only four sound channels, only four of the sampled sounds can be heard at once. This limitation is not as bad as it sounds because, during normal play, a drummer very rarely hits more than three drums simultaneously.

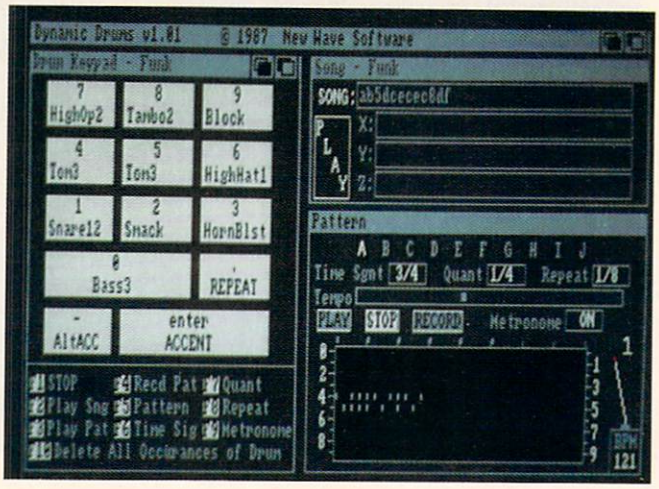
Time-out for a brief discussion of the included drum samples. First of all, they are superb. I was particularly impressed with the bass drum and snare samples. The tom samples are great, and New Wave also did a good job of capturing the cymbal sound, an instrument with an extremely complex waveform.

New Wave Software has chosen not only to supply quality samples, but to provide quantity as well. The program comes with over one hundred samples! In addition to the standard drumkit which includes bass drum, snare, tom, high-hat, and cymbal samples, other percussion sounds like bongos, woodblocks, hand claps, and cowbells are also included. To spice up your riffs, non-percussion instruments are included, like bass and electric guitar, electric piano, and a horn blast. Finally, a set of novelty sounds (such as breaking glass) can add some fun to your playing.

(continued)

At the time of this review, New Wave Software was putting together another disk of approximately 75–100 more sampled sounds for release in the near future. I was also delighted to learn that the program reads samples recorded in the FutureSound format. This flexibility means that not only are the samples smaller than the standard IFF disk hogs, but the user can record his own samples with the FutureSound digitizer for use with Dynamic Drums. The program has very limited success in reading IFF format samples—this is one of my few qualms about Dynamic Drums. I would love to be able to load Instant Music and other EA samples. New Wave software is aware of this problem, and they are working on it.

The next step on the journey to percussion paradise is the construction of a "pattern." The term pattern, common to most drum machines, is used to describe which drum is played at which time during a measure. This representation is similar to the five staff method of writing melodies.



In Dynamic Drums, there are basically two different methods of writing patterns. The first method involves manually editing a graphic display of the pattern. This display represents a graph with the number of the drum sample on the y-axis and the sub-division of a measure on the x-axis. The mouse is used to select the drum beat by clicking on the appropriate spot on the "grid." For the beginning user, however, finding the right place to click can be difficult. The standard mouse arrow pointer does not give a clear indication of where the "hot-spot" is in relation to the small area that needs to be clicked. After a bit of practice, this nuisance virtually disappears. As the manual suggests, this method is much better for editing an existing pattern.

The second pattern recording method is a bit easier. Selecting a box marked "record" sets the program's real power in motion. When the numeric keys are pressed, the played sequence is recorded. If the bass drum key is pressed four times, evenly spaced in a measure (once every quarter note), the program will continue to play a steady bass drum beat. Pressing more keys will layer the various sounds over the existing pattern. If you add two or three more drum sounds, a very funky drum pattern comes into being.

Various editing keys, like "repeat" and "delete all occurrences of drum," help speed up the creation process. A visual and aural metronome can be invoked to help with timing. For even more assistance, Dynamic Drums offers a unique feature called "quantizing" which automatically corrects errors. The user can set the program to round out the timing of recorded beats, so that no matter how off-beat the keypad is pressed, the program will record the beats in perfect sync. Quantizing allows users with no prior percussion experience to create professional sounding patterns. The finished pattern can then be heard by selecting the "play" box.

The tempo of recording and playback can be adjusted with a sliding gadget. Dynamic Drums also offers the ability to sync the tempo with an external MIDI source. The speed is measured in beats per minute—musicians will appreciate this touch. The ability to play the keypad even while a pattern is playing is one of my favorite features of this program. This feature allows the user to "jam" with the pattern. You can save the pattern to disk if you wish. Several good demo patterns are also included with the package.

Dynamic Drums can store ten patterns in memory, in pattern banks A through J. These ten patterns can be strung together to create a "song." Songs are specified using the letter of the pattern bank, together with the number of repetitions of that pattern. For instance, "3ab2c" would play pattern "a" three times, then pattern "b", and then pattern "c" twice. This uncomplicated approach makes it very easy to write a extended song, complete with drum fills.

Features abound in this program. For example, Dynamics Drums allows you to change the volume and tone of each drum sample in memory. An alternate tuning/volume can be assigned to each sample. This alternate sound can be accessed by using the "accent" key like a shift key. This feature makes twenty sounds available to the user at any time. Alternate tunings are especially effective on tuned percussion, like toms and bongos.

Another unique feature of this program is called "randomizing." Randomizing makes a pattern sound more realistic by simulating the slight change in pitch a drum makes as the

stick hits it in different places on the head. When this feature is selected, the program randomly picks a pitch between the pitch of the current setting and its accented setting. This revision makes the pattern sound less mechanical than the usual sound from a drum machine.

Though not implemented in the standard package, Dynamic Drums allows your Amiga to be connected to the Casio DP-1 drum pads (retail approx. $60). The drummer can then play the pads in live mode, playing any four samples like real drums. Contact New Wave software for more details on this.

My final comment concerning Dynamic Drums revolves not around the program itself, but the people behind it. With all programs, good customer support is essential. The people at New Wave Software must require chiropractors because they bend over backwards to guarantee customer satisfaction. During the course of this review, they were most courteous in responding to my questions and suggestions, even going as far as having the actual programmer return my call. A few suggestions I made will be implemented in an upcoming update. This type of response from a company to customers delights me.

It is probably apparent by now that I really like this program. I am absolutely tickled by the fact that one program, at less than one-tenth of the price of others, outperforms the professional drum machine I nearly purchased. The program worked flawlessly—not a single crash in all the hours I spent with it.

New Wave Software really impressed me with their first Amiga product (Their second, a professional sixteen track sequencer program with drum machine built in, is due out soon.). If you are an Amiga user with any percussion background, or if you would like to build such a background, I recommend you run to your friendly neighborhood software dealer and get a set of Dynamic Drums.

•AC•

Dynamic Drums
List Price: $79.95

New Wave Software
P.O. Box 438
St. Clair Shores, MI 48040

15-Puzzle

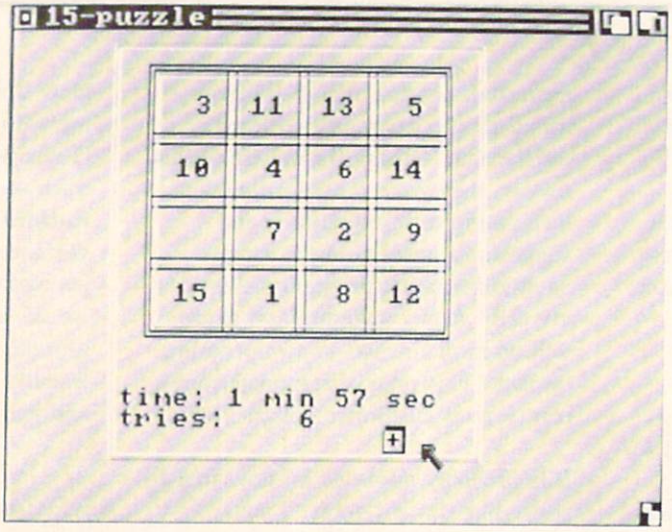
in AmigaBASIC™

The Classic Puzzle Comes to the Amiga

by Zoltan Szepesi

The 15-Puzzle was invented in 1878 by an Englishman named Sam Lloyd. This new invention caused much excitement, from the general population and mathematicians alike. The reaction was similar to the fury RUBIK's Cube created some years ago.

15-Puzzle begins by showing a 4x4 square board filled with the numbers 1 to 15 randomly scrambled and one empty square. The object is to put the numbers in numerical order using the following method: Any number adjacent to the empty square can be moved into the empty square, thus leaving its former place empty for another number to move into.



This program was made for the Amiga in AmigaBASIC. Other similar programs can be found for the AMIGA that use alphabetic letters, instead of numbers. One such program was created by Bill Boegelin and listed in *COMPUTE!* magazine (p.79–81, May 1986).

We use numbers in the program presented here, thus making the game easier to follow. We also use randomized distribution of numbers to set up the board, instead of scrambling the original, ordered distribution.

Randomly selecting the numbers, however, can result in a distribution which cannot be solved. There are nearly 21 trillion (For Europeans, this is 21 billion.) possible starting distributions of 16 numbers (more precisely, 16!=16x15x14x13x12x11x10x9x8x7x6x5x4x3x2x1). Mathematicians have proven that half these distributions cannot be solved There is a simple, fast calculation for checking whether a set can be solved or not. We will describe it later.

The Program

After setting up a low resolution screen and choosing four colors, a large window is opened, asking if you want instructions. The Start routine follows, which directs the different steps of the game.

The Init routine defines the "times" function and contains the Selection and Verify subroutines. Selection creates a randomly distributed array, a(v), of the first 16 numbers. Since half these distributions are unsolvable, the Verify subroutine checks if the chosen setup is good or not. If the problem cannot be solved, the program returns to the Selection routine and creates another distribution of numbers.

According to D.D. Spencer's *Game Playing with Computers*, a simple rule is derived for the checking done above. This rule is as follows:

From the first to the fifteenth position, count how many numbers are of lower value than the number in the starting position (count the empty space as 16) and add them together. Add 1 to the sum, if the empty square is in one of these positions:

2,4,5,7,10,12,13 or 15.

(continued on page 52)

Amiga is a trademark of Commodore-Amiga, Inc.

AMIGA Dealers!

We are a Dallas Distributor exporting to customers in over 20 Nations around the globe.
Call us today at 1-800-531-4747 (Overseas call 214-6601763) and ask for your Dealer Kit !
Here is just an overview of some of the AMIGA products we stock.

Shakespeare

The first color Desktop Publishing Program for the Amiga. Works in any resolution. Powerful features for Textformating and page layout. True multitasking software. Not copy protected.

Super Saver of the Month !

City Desk

Plutos

Fast -shoot'em up- style game.
Very good sound effects;
High resolution-smooth scrolling graphics

Opcode systems

Music Mouse

One of the most advanced combinations of artificial intelligence and randomized sound generation available for the Amiga today ! Very good and incredible fast Software.

DigiTek, Inc.

Hollywood Poker is the hottest Strip-poker for the Amiga today ...!
Give customers a lot of Party fun.

Drum Studio is the ultimate toolbox to create drum sounds !
Start rockin' at a reasonable price.

Hottest Game of the Month !

Arkanoid

Arcade-style smooth graphics; excellent sound and super speed !

Flicker Master
Flicker Master Flicker Master
Flicker Master

Reduces Interlace Flicker dramatically.
For the price conscious customer who doesn't want to buy a high persistence monitor.

Texture

This is the AMIGA-Version of a true professional music editor first released for MS-DOS machines and the MAC. As midi Texture requires a Roland MPU-401 interface and the MIF-AMG adapter.

InterComputing Inc.

2100 N. Hwy 360, Suite 2101
Grand Prairie, TX 75050-1015
Order line : 800-531-4747
Technical Support : 214-988-3500
International calls : 214-660-1763
Telex # : 4932133 ICINCUI
FAX : 214-660 3695

We stock a full line of Amiga soft and hardware since 1985, when the machine was first introduced, and serve customers in more than 20 Nations around the world. Most of our employees own an Amiga and use the available products.
We may be new to you, but we're experienced with the Amiga !
Check our prices, and you'll see that we try to be quite fair. Good products are worth the money, and you will not be disappointed with our service.
We are a member of the Dallas Better Business Bureau.

Barbarian

A must for your customer's game collection !
The graphics are simply fantastic...

the MICROFICHE filer

Unique Database System for Text and IFF Pictures. Very intuitive and easy to use !

Butcher

Very powerful conversion & processing utility for digitized graphics !

Starboard2 Memory cards:

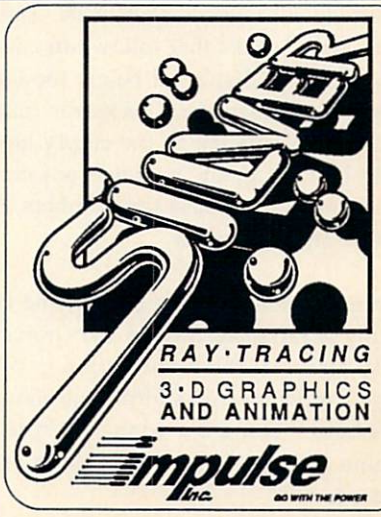
Microbotics

Versions for the A500
A1000
and A2000

Sculpt 3D

Extremely easy to use 3D-solid modeling Program. Uses RAY-TRACING techniques to create incredible computer graphics one simply couldn't just 'paint' . . . !

and more (please call for a complete catalog)



If the sum is even, the puzzle can be solved. If the result is odd, it cannot be solved.

Once a solvable setup is found, the Drawboard subroutine forms the board. The Drawscreen routine then places the numbers in the 16 squares, after the one dimensional a(v) array is transformed to a two dimensional c$(x,y) array. This change is necessary to enable the mouse for changing, the positions of the numbers.

After the numbers are drawn, the Amiga says, "ready," and the Play subroutine takes over. The timer starts, and the number of clicks that follow are counted. Both these numbers are displayed below the board. When the mouse clicks a number, the Checkerror routine checks if the chosen number is adjacent to the empty block (If it is not, you are told to "Try again."). If it is adjacent, then the two blocks are interchanged and the numbers are checked to see if they are in the final order.

To make this checking simple, the two dimensional c$(x,y) array is retransformed to the equivalent one dimensional a(v) array. If the numbers are in order, the starting four notes of Beethoven's Fifth Symphony resound and AMIGA declares, "You are a winner." With the More routine, a new game can be initiated by clicking on the + sign below the board.

Three simple subprograms (Talk(a$), Position(x,y,a$) and Beethoven) are used in different parts of the program.

Conclusions

The difficulty in solving the 15-Puzzle depends on the particular order of the numbers. If you are new at the game, you will probably need more than five minutes and 100+ steps to finish. After some practice, you should get down to less than 100 steps.

With the Amiga mouse, play is more than twice as fast as keying in the numbers, as in a similar program for the C-64 computer. The C-64 version can be downloaded from the Quantumlink BBS, where it was placed in a 10 most popular programs list last September.

```
REM Change to 80 column screen, please, if it is not set.
CLEAR,10000,6000
'15-puzzle_v3.8 (8-22-87)
'By Z.Szepesi,2611 Saybrook Dr.,Pittsburgh,PA 15235-5131
REM Should have an 80 column screen!
SCREEN 1,300,200,2,1
DEFINT a-z:DIM a(16),c(3,3),c$(3,3),conf(3,3,1)
DEF FNmin=((TIMER-starttime!)\60)
DEF FNsec=((TIMER-starttime!) MOD 60)
WINDOW 1,"15-puzzle",,,1
PALETTE 0,0,1,.6:PALETTE 1,0,0,0
PALETTE 2,0,1,1:PALETTE 3,0,0,1
COLOR 1,0
INPUT"*Do you want Instructions (y/n)";q$
IF q$="y" THEN GOSUB instructions
'*********************************************************
start:
CLS:PRINT "            PLEASE WAIT."
GOSUB Init
CLS:GOSUB Drawboard
GOSUB Drawscreen
   Talk "ready."
   starttime!=TIMER
WHILE WINDOW(7)<>0
   GOSUB Play
WEND
Done:
BEEP:WINDOW CLOSE 1
END
'*********************************************************
Init:
Talk " "
tries=0:ok=0:RANDOMIZE TIMER
Selection:'*****************************************
FOR v=1 TO 16:a(v)=0:NEXT
FOR i=1 TO 16
10 v=INT((16*RND)+1):IF a(v) THEN 10
a(v)=i:NEXT
Verify: '*******************************************
su=0
FOR i=1 TO 15:FOR j=i+1 TO 16
IF a(i)>a(j) THEN su=su+1
NEXT j:NEXT i
RESTORE:FOR i=1 TO 8:READ v1:IF a(v1)=16 THEN su=su+1
NEXT i
DATA 2,4,5,7,10,12,13,15
IF su MOD 2 THEN Selection
RETURN
'*********************************************************
Drawboard:
   LINE (46,5)-(210,162),3,b
   LINE (45,4)-(211,163),3,b
FOR y=0 TO 3:FOR x=0 TO 3
   x1=64+32*x:y1=16+24*y
   LINE (x1,y1)-(x1+32,y1+24),3,b
```

```
   LINE (x1-2,y1-2)-(32+2+x1,24+2+y1),1,b
   LINE (60,12)-(64+4+4*32,16+4+4*24),1,b
   morex=167:morey=150
   LINE (morex,morey)-(morex+10,morey+10),1,bf
   conf(x,y,0)=x1:conf(x,y,1)=y1
NEXT x:NEXT y:RETURN
Drawscreen: '*********************************************
v=1
FOR y=0 TO 3:FOR x=0 TO 3
   c(x,y)=a(v):c$(x,y)=RIGHT$(STR$(a(v)),2)
      IF a(v)=16 THEN blankx=x:blanky=y:c$(blankx,blanky)="  "
   Position (x+1)*4+6,(y+1)*3+1,c$(x,y)
   v=v+1
NEXT x:NEXT y
   Position 7,18,"time:          "
   Position 7,19,"tries:   0     "
   Position 22,20,"+"' new game gadget
RETURN
'*********************************************************
Play:
   LOCATE 18,12:PRINT FNmin;"min";FNsec;"sec "
WHILE MOUSE(0)
   mousex=MOUSE(3):mousey=MOUSE(4)
      FOR y=0 TO 3:FOR x=0 TO 3
        IF (mousex>conf(x,y,0) AND mousex<conf(x,y,0)+32) AND
(mousey>conf(x,y,1) AND mousey<conf(x,y,1)+25) THEN GOSUB
Checkerror:RETURN
      NEXT x:NEXT y
   GOSUB More
WEND:RETURN
'*********************************************************
Checkerror:
IF(ABS(x-blankx)>1 OR ABS(y-blanky)>1) OR ((x<>blankx AND
y<>blanky)) THEN
   Talk "try again." 'error
ELSE 'no error
   SWAP c$(x,y),c$(blankx,blanky):c(x,y)=VAL(c$(x,y))
   Position (x+1)*4+6,(y+1)*3+1,c$(x,y)
   SWAP x,blankx:SWAP y,blanky
   Position (x+1)*4+6,(y+1)*3+1,c$(x,y)
END IF
tries=tries+1
Position 16,19,STR$(tries)
WHILE MOUSE(0):WEND
'Check for a win*********************************************
FOR y=0 TO 3:FOR x=0 TO 3
   i=x+4*y+1:a(i)=VAL(c$(x,y))
      IF c$(x,y)="  " THEN a(i)=16
      IF a(i)<>i THEN RETURN
NEXT x:NEXT y
ok=1:CALL Beethoven:FOR z=1 TO 15000:NEXT z
Talk "Congratulations. You are a winner."
'*********************************************************
More:
WHILE MOUSE(0) OR ok 'Another game?
   mousex=MOUSE(3):mousey=MOUSE(4)
      IF MOUSE(0)=0 AND (mousex>morex AND mousex<morex+10) AND
(mousey>morey AND mousey<morey+10) THEN GOTO start
      IF WINDOW(7)=0 THEN Done
WEND:RETURN
'*********************************************************
instructions:
PRINT
PRINT "**The object of the game is"
PRINT "to move the numbers around so that"
PRINT "they are in order from 1 to 15."
PRINT
PRINT "  A move is made by clicking the"
PRINT "left mouse button on the number you"
PRINT "want to move. The number to be"
PRINT "moved must be adjacent to the empty"
PRINT "square. The clicked number then"
PRINT "moves to the empty square."
PRINT
PRINT " You win, when the board looks"
PRINT "like the figure that follows."
PRINT
PRINT "**Press any key to continue"
WHILE INKEY$ = "":WEND:CLS
FOR i = 1 TO 16:a(i)=i:NEXT i
GOSUB Drawboard
GOSUB Drawscreen
PRINT:PRINT "**Press any key to continue"
WHILE INKEY$ = "":WEND
CLS:RETURN

SUB Talk(a$) STATIC
SAY TRANSLATE$(a$)
END SUB

SUB Position(x,y,a$) STATIC
LOCATE y,x:PRINT a$
END SUB

SUB Beethoven STATIC
t=4:RESTORE 100
FOR i=1 TO 10
READ d,v
FOR s=0 TO 3
SOUND 400,1,0,s
READ f
SOUND f,d*t,v,s
SOUND 400,1,0,s
NEXT s:NEXT i
100 DATA 1,0,400,400,400,400
DATA 1,150,195.998,391.995,783.991,1567.982
DATA 1,150,195.998,391.995,783.991,1567.982
DATA 1,150,195.998,391.995,783.991,1567.982
DATA 4,150,155.563,311.127,622.254,1244.508
DATA 1,0,400,400,400,400
DATA 1,150,174.614,349.228,698.456,1396.913
DATA 1,150,174.614,349.228,698.456,1396.913
DATA 1,150,174.614,349.228,698.456,1396.913
DATA 8,150,146.832,293.665,587.330,1174.659
END SUB
```

•AC•

WordPerfect

by Steve Hull

GEnie: LightRaider
People Link: St.Ephen

A year ago last spring, it had the sound of just one more of the many wild rumors circulating about the Amiga: "Hey, didja hear? Ashton-Tate's porting Dbase III to the Amiga." "I just heard Lotus is working on 1–2–3 for the Amiga." "Did you hear there's going to be an Amiga version of WordPerfect?"

Yeah, right, I thought. And if my grandmother had a blitter she could multitask.

To me, WordPerfect for the Amiga had a distinct too-good-to-be-true ring to it. I had been using the MS-DOS version of WordPerfect as a word processor, directory utility, and program text editor for over a year, and it was one of the few things having to do with the IBM that impressed me. Here was a program that consistently earned superlatives in the toughest product reviews. In power, ease of use, customer support, and documentation, WordPerfect had everyone else playing follow-the-leader. In short, a class act. The kind of class act I didn't expect to see translated to a machine that, at the time, had an installed user base of a little over 100,000.

Yet "WordPerfect for the Amiga" was a rumor with unusual staying power, and in September of 1986, WordPerfect Corporation made it official: An Amiga version was in the works. At the Las Vegas COMDEX that fall, WordPerfect fielded a booth staffed by public relations personnel and key members of the software design team—plus a working beta copy.

As is often said—seeing is believing. But even as I type this review, with WordPerfect virtually under my fingers, it's one of those most fortuitous circumstances that takes a while to fully sink in.

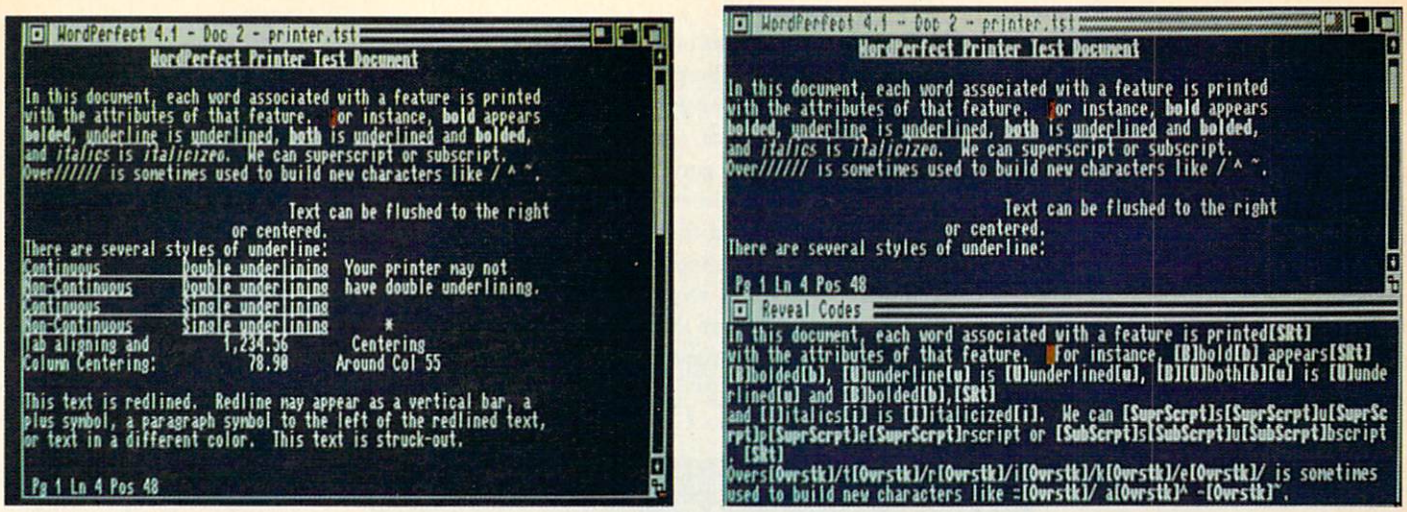
What's The Big Deal?

Perhaps you have been sitting off by the sidelines the past year, while us excitable writer-types eagerly (and vocally) anticipated WordPerfect's release, finding it all a little hard to understand. How is it that a word processor, of all things, got to be the most anxiously-awaited title on a graphics-and-sound magic-box like the Amiga? And how could anyone ask $395 for a word processor?

For someone who doesn't do a lot of word processing, there may be no way to answer that question. Yet you don't have to be John Updike to understand—and even appreciate—why WordPerfect is on the verge of becoming the widest cross-system standard the software industry has ever produced.

A Summary of Strengths

Historically, WordPerfect's greatest strength has been its usefulness across a vast range of word processing applications. To the casual user, WordPerfect presents a clean, uncluttered screen and a logical user interface. The jump from typewriter to word processor is an easy one, even for the amateur—just sit down and type. There is no WordStar-like need to memorize a plethora of thoroughly illogical control-key combinations to effect even the simplest functions. Need to back up, or go back a line? The arrow keys move the cursor just as you would expect. To save a document, press the function key marked SAVE. To print, press PRINT. It doesn't get much easier than that.



"On the left, WordPerfect's Printer.TST file as it appears on the edit screen. Many of the effects, such as super/subscripts, justification and overstrike, do not appear on screen. On the right, the same screen examined with Reveal Codes. Special WordPerfect formatting codes appear boldfaced in brackets."

WordPerfect does not sacrifice power for ease of use. It offers advanced formatting features, such as footnotes and endnotes, decimal tab alignment, and multiple columns on screen. On-line facilities allow you to create tables of contents, indexes, and outlines. One of the smartest spell-checkers in the business is at your fingertips, along with a thesaurus. WordPerfect performs math functions on tables of numbers in a document, and its mail merge works better than many databases. Its macro functions are so powerful that, in my office, I have reduced a half hour of tedious manual formatting to thirty seconds and *two keystrokes*. Best of all, each feature is easy to learn, so you can learn as little or as much as you need—WordPerfect's usefulness grows as you grow.

Compatibility Plus

One of the most exciting things about WordPerfect for the Amiga is WordPerfect Corporation's commitment to keystroke- and file-compatibility—as far as is possible—across different machine systems (This can also be a two-edged sword; more on that later.).

If you learned WordPerfect on an MS-DOS system, keystroke compatibility means you'll be right at home with the Amiga version. With a few exceptions, the two versions operate identically. Pressing F5 brings up a list of available files on both versions (though the displays themselves are markedly different); likewise, Shift-F7 calls the PRINT menu, ALT-right arrow moves the cursor to the end of the current line, and so on. The transition between "home" and "work" machines is nearly effortless.

Transition between machine systems is also facilitated by close attention to file compatibility. This means that while you can't, for instance, directly access an MS-DOS format floppy disk using the Amiga version of WordPerfect, the codes used to represent the various functions—underlining, margins, and so on—are identical. Once you transfer the MS-DOS file to an Amiga disk (over phone lines or using the PCTools utilities included on the Amiga Extras disk), you can load the file created by the MS-DOS WordPerfect into Amiga WordPerfect and pick up writing where you left off. When you consider that WordPerfect is now available for the Amiga, Atari ST, IBM-PC, VAX, Data General, and Apple II (with UNIX and Macintosh versions currently in development), it becomes clear that WordPerfect is fast becoming a member of one of the rarest breeds: a true cross-system standard.

WordPerfect With Intuition

One of the fears I had when I first heard of WordPerfect Corporation's assurance of keystroke-compatibility was that Amiga users would lose the use of the Intuition features. I needn't have worried. The Amiga version of WordPerfect supports the full Intuition interface, including multiple windows and multi-tasking. Pull-down menu options duplicate those functions accessed through the standard WordPerfect key combinations. The mouse is an especially welcome addition; you can use it to instantly place the cursor at any point in the screen, or "paint" portions of the document for special formatting.

Out Of The Box

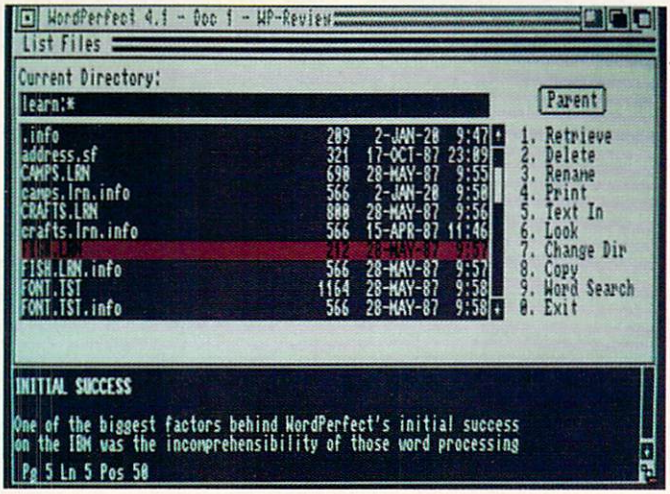
WordPerfect is packaged in a sturdy cloth-covered cardboard slipcase. Two function-key templates are included: one for the Amiga 500 and 1000, and one for the Amiga 2000. Don't lose

(continued)

'em; in spite of WordPerfect's excellent on-line help facility, you will find yourself referring to the template often until you become familiar with the program. A color-coded quick reference card is also included.

The program covers four disks, though only one is needed for most routine operations. A set of transparent pressure-sensitive decals is also included. These decals relabel the numeric keypad on the Amiga 1000 with the standard IBM-PC functions of PgUp, PgDn, Ins, Home, End, plus arrows. If you have already applied the decals that come with the Transformer package, you won't need to relabel for WordPerfect.

The WordPerfect manual is outstanding. It was rewritten top-to-bottom for the Amiga, and like the program, well serves the needs of novice and professional alike. It features a comprehensive "Getting Started" section, a 200-page, 29-lesson tutorial, an exhaustive reference section, several technical appendices, a glossary of terms, and a full index.



"WordPerfect's List Files screen allows many file transfer options. In addition, its Word Search feature searches for keywords within files."

Disk Drive Requirements

I must address WordPerfect Corporation's statement that Amiga WordPerfect may be adequately operated with only one disk drive. I suppose whether or not this is true is a highly subjective matter—however, one-drive operation is addressed nowhere in the documentation. Step-by-step installation instructions are included for hard drive systems and dual-floppy configurations, but the single-drive user is left out.

Because of the way the Amiga's operating system was designed, it is technically possible to run almost any program with only one disk drive—Intuition simply prompts a disk-swap when required. The problem with WordPerfect is, between its timed backups (if desired), stand-alone printing program, and many overlay files in the LIBS directory, you pretty much need to keep the WordPerfect system disk in the internal drive at all times. Operations requiring an additional disk—such as the thesaurus and spelling checker—require additional swapping. The most extreme case I encountered was initial printer selection. With two disks, the process was simple. With one disk, the process required *44* disk swaps! Admittedly, this is not something the average user will have to do often—but when he does, it will be a memorable experience.

Don't misunderstand my point; this is not a criticism of the program. It is not unreasonable for a program of WordPerfect's power to require an enhanced configuration. You can also run the program with one floppy drive (and some patience). I only wish WordPerfect Corporation had been more straightforward in their packaging.

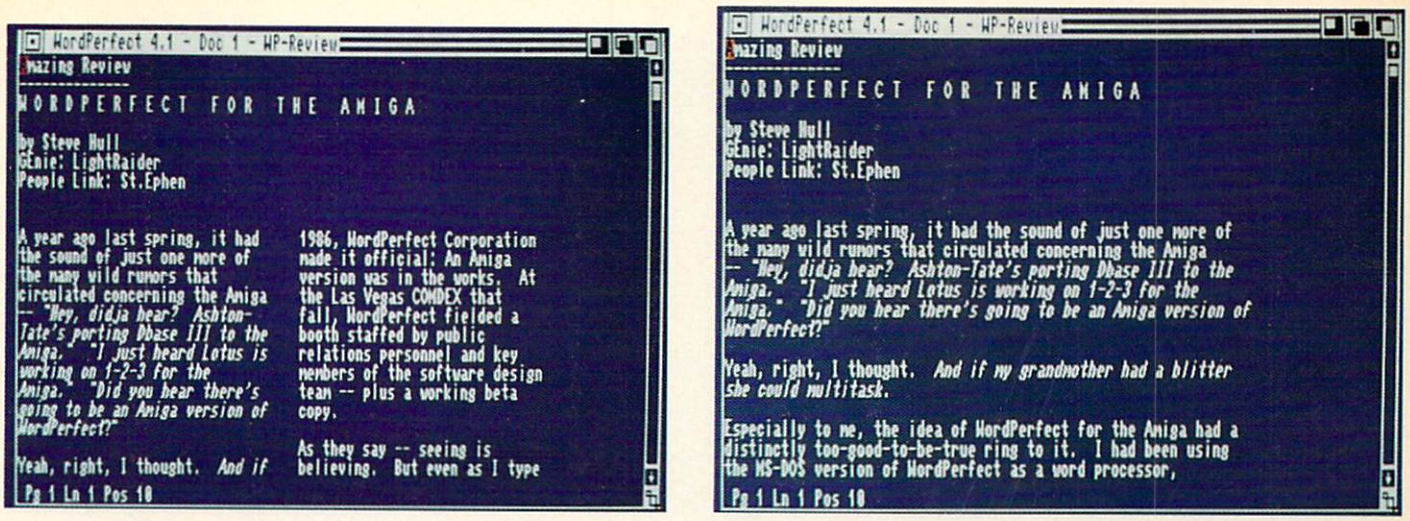
Fully Featured

I don't want to spend a lot of time in this review with the more mundane aspects of word processing, such as hanging indents, right-justification, insert or typeover mode, *et cetera*. If you're wondering if WordPerfect supports a specific standard word processing function, I can assure you that it probably does. This is not as flippant a remark as it appears—in a recent comparison chart that pitted WordPerfect against six other MS-DOS heavyweights, the program missed only eight out of a possible 132 listed word processing features (By comparison, Wordstar 2000 Plus missed 29.). Some of the features WordPerfect got dinged for missing were esoteric indeed, such as a telecomm module. I won't take up any more space assuring you of WordPerfect's ability to do centering, boldface, or cut-and-paste. Trust me. It can.

Rather than concentrate on the basics (that even Notepad can handle, after a fashion), let's explore the features that separate WordPerfect from its competition. If you're wondering whether I am a publicist getting paid on the sly by WordPerfect Corporation, I'll spend some time on those imperfect areas of WordPerfect. As a matter of fact, that's not a bad place to start.

Initial Success

One of the biggest factors behind WordPerfect's initial success on the IBM was the incomprehensibility of the word processing programs that preceded it. The most widely-used were, for the most part, ports of popular CP/M programs, and carried much of the awkward user interface early microcomputer users had to accept to get power out of such limited architecture. For instance, if



"Multiple columns are easily accomplished. You may write a document in multiple-column format, or reformat previously written text (right)."

you wanted to boldface a word, previous programs required you to bracket the word with control codes actually imbedded within the text on screen. Likewise, on-screen codes were needed to underline, set line spacing and set margins. People became accustomed to it, but I never met anyone who loved it.

Enter WordPerfect, with as close to WYSIWYG (What You See Is What You Get) as anyone had ever seen on a microcomputer. WordPerfect used those same codes, but most of them were tied to simple function-key operations. Rather than having to remember cryptic control codes to initiate boldfacing, WordPerfect users, for example, merely pressed the F6 key (marked BOLD on the template) where they wanted boldfacing to begin, typed the boldfaced word, then F6 again to end boldfacing. The boldfaced portion appeared in a different color than the normal text, and, best of all, the codes that made it all happen were tucked away "under" the document (more on that in a minute)—on screen were only the typed words. Margins appeared as they would print (within screen constraints), and double-spaced text appeared double-spaced on screen. It was quite a revolution at the time.

Amiga users started from a radically different "baseline." The very first word processing software written for the Amiga supported true on-screen boldface—plus italics, underlining, and, in some cases, multiple fonts. The fact that Amiga WordPerfect supports these attributes isn't news; it's business as usual.

So even though Amiga WordPerfect's WYSIWYG is far superior to its record-breaking MS-DOS cousin, the term becomes quite relative when compared to features that are quickly becoming standard fare for Amiga word processing programs. Amiga WordPerfect accurately displays bold, underline, italics, margins, line spacing, tabs, and centering on line. Its display does not reflect right-justification, super- or subscripts, centering on page, red-lined text, overstrikes, or differences between pica, elite, condensed or expanded text. Font selection is limited to your printer's stock fonts, and if you have a color printer, you'd better be a whiz at control codes to use color in your documents. Graphics, of course, are out of the question.

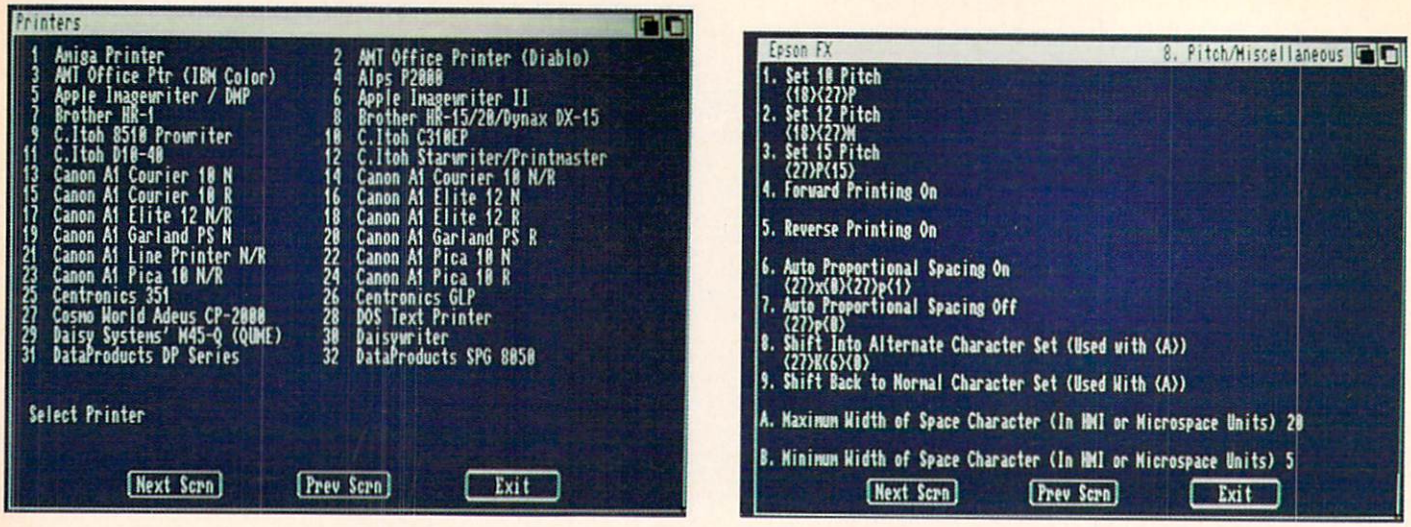
This, then, is the dark side of compatibility. While it's great to be able to use files composed on the MS-DOS version, it also means future revisions of WordPerfect will tend to stay within the limits of the lowest common hardware denominator. For the MS-DOS crowd, that's hardly a limitation. For Amiga users, WordPerfect's screen display is already a generation behind.

Secret Codes

I've already mentioned that WordPerfect hid its formatting codes "under" the document. To be more precise, the program imbeds the codes within the document at the point where they are to take effect. While the codes are not normally visible, they can be "revealed" and examined.

Pressing Alt-F3 activates a function known as Reveal Codes. This action opens a separate window below the document display, showing a mnemonic representation of each attribute where it occurs in the document. The simplest document will have the code *[SRt]*—soft return—at most line-breaks; *[HRt]* denotes a hard, or "forced," carriage return. An underlined phrase appears bracketed by the codes *[U]* for underlining on and *[u]* for underlining off. Most codes are easy to understand: *[C]* designates a centered line; *[B]* signifies boldface; and I'll leave it up to you to figure out what *[TAB]* means. Other codes are less obvious: an *[A]* indicates a flush-right

(continued)



"On the left, 32 of the 250 printers directly supported by WordPerfect. Each printer's definition may be examined and edited using WP's PrintDef program (right)."

line. Super- or sub-scripting really looks strange under Reveal Codes. Apparently, WordPerfect formats each letter separately. The word "super," marked for superscripts, appears as *[SuprScrpt]s[SuprScrpt]u[SuprScrpt]p [SuprScrpt]e[SuprScrpt]r.*

Once in Reveal Codes, you can navigate through the document using most of the standard methods (cursor keys, PgUp or PgDn). A proportional slider is available as well, but it doesn't work in Reveal Codes mode. The advantage to Reveal Codes is that you can move the cursor through the formatting codes one at a time, deleting as necessary.

WordPerfect's handing of formatting codes is a strength, but it does require some getting used to. It takes a while for the typical beginner to realize, for instance, that even though he deleted an italicized word, the *codes* for italics may still be there, "under" the document. In practice, this problem would be minor; should a word be inadvertently inserted between the hidden codes, the italics on screen would make the mistake immediately evident. If, on the other hand, the forgotten code happened to be Strike-out, the first indication that something was amiss would show up when the printer began pumping out a whole document with dashes overprinting the text. For this reason, one disgruntled user gave WordPerfect the thumbs-down for failing a test he termed WYGINS (What You Get Is No Surprise).

Word Processing Powerhouse

WordPerfect has its idiosyncrasies, but, by and large, these are products of the normal learning curve. The fact remains that most users who opt for WordPerfect do so not because of its screen display, but because the program is quite simply a word processing powerhouse. It not only covers the common word processing functions, but it does so exceptionally well. Let me give you some examples.

The Editor's Desk

Composing a document using WordPerfect is simple. Word wrap to the next line is automatic, and margins and line spacing appear as they will print. You may toggle freely between insert and overstrike modes. WordPerfect redefines the Amiga numeric keypad to act like an IBM keypad, though you may revert back to numeric entry if you choose. With the keypad redefined, you can move the cursor through a document a letter, line, or page at a time. A proportional scroll gadget allows you to do nearly the same thing with the mouse.

The most common word processing tasks (forward search, italics, left-indent, List Files, bold, exit, underline, and save documents) may be accessed directly from the function keys. Pressing F1 alone cancels or undoes many operations.

Pressing shift keys in conjunction with function keys is less routine, but still relatively common: super/subscripts, backwards search, opening or switching between multiple document windows, left/right indent, insert date, center on line, print, access line formatting options, and retrieve documents.

Pressing the ALT key and function keys together summons more advanced features—the thesaurus, replace, Reveal Codes, Block, Mark Text, flush right, summon math/multiple columns options, page format options, and macros.

Pressing the CTRL key along with the function keys opens a new CLI window, activates the spelling checker,

offers screen options, text import/export/lock, tab align, footnotes or endnotes, printing formats, and Macro Define.

Function key F9 is reserved for use with WordPerfect's mail merge functions.

Besides the function keys, WordPerfect also combines the Amiga's standard keys and the redefined keypad to effect different functions: CTRL-RETURN forces a page break; CTRL-right or -left arrow moves the cursor to the right or left one word at a time; HOME-HOME-down arrow moves to the end of the document. In case you forget any of this, pressing the HELP key brings up an on-line help facility cross-referenced to functions and keystroke combinations!

Now that I've given you a quick overview of WordPerfect's capabilities, let's take a look at some of the more interesting ones in depth.

Search ...

WordPerfect's Search function offers much flexibility. Searches may be conducted either forward or backward through the text, case-sensitive or not. Lowercase search criteria will yield upper- or lowercase matches; uppercase letters specified in a search will match only uppercase. Therefore, a search for the string "test" will find "test," "Test," and "TEST." Searching on "Test" will find "Test" and "TEST." Searching on "TEST" will produce only the case-for-case match, "TEST." To limit searches to whole words (Find "test" and ignore "attest" and "testify."), you must delimit the search criteria with a blank space at each end. In addition to words and phrases, you may also include hidden formatting codes in search criteria.

Amiga WordPerfect omits one very useful search facility implemented in the MS-DOS version: the ability to substitute wildcard characters in search criteria.

... And Replace

When combined with WordPerfect's search abilities, Replace becomes doubly powerful. The Replace criteria follow the same guidelines as search criteria and may be executed globally by document, within the limits of a defined block of text, or with verification.

Theoretically, WordPerfect's ability to include hidden formatting codes in search and replace strings is a real plus. In operation, the function suffers from an occasional glitch.

As an experiment, I changed all occurrences of the name, "WordPerfect" in this review from normal text to boldface. The operation worked like a charm. Intrigued, I decided to try something a little trickier—changing my now-boldfaced "WordPerfect's" to underlined and italicized. *Voila!* It worked again!

Delighted with my own cleverness, I then set to return the *"WordPerfect"* strings back to normal text—and that's where I hit the snag. After one completely unsuccessful search operation (I mistakenly reversed the underline and italics codes.), I carefully matched my search criteria against what I saw in the Reveal Codes window and tried again.

This time I got "WordPerfect" in my document from top-to-bottom. That's not a typo—the underlining begins under the "o." Strangest of all, nothing I tried using Search and Replace would restore the odd strings to normal text. Search would locate the criteria, but Replace wouldn't replace. I finally had to go back and remove the spurious underlining manually.

(continued)

File Handling

File storage and retrieval aren't usually operations that make the heart beat more quickly, but WordPerfect's options are something special. Aside from handling its own custom format, WordPerfect also allows you to save and recall files as unformatted ASCII text, making WordPerfect an ideal text editor for programmers.

If you attempt to load a non-WordPerfect document using the standard Retrieve function, the program informs you of the discrepancy and allows you to either cancel or continue. I was very pleased to see, at least in the case of my old Scribble!-formatted documents, that WordPerfect simply stripped out the formatting characters it didn't recognize, leaving straight text requiring minimum corrective reformatting.

WordPerfect allows you to "lock" documents using a sophisticated encryption technique. Each locked file may have its own password, up to 75 letters long.

WordPerfect takes several precautions to safeguard your files. You may configure the program to automatically backup the document you're editing at specified time intervals; you may also set the program to save an unedited backup version of your document each time you perform a "save."

DIRUTIL-WP

While we're on the subject of file handling, WordPerfect's List Files function deserves special mention. F5 calls up a special List Files submenu with many features you'll wonder how you ever did without. Besides the standard options of retrieve (WordPerfect-format documents or unformatted text), delete, and rename, List Files also allows you to change directories or simply examine their contents, print any file listed without loading it into the editor, and copy files from one source to another. You may also scan the contents of any file without performing an actual retrieve.

An especially intriguing feature included under List Files is Word Search, whereby you may instruct the program to search a directory for specific words or phrases. This is not a simple title search (though WordPerfect has a way of doing that too), but a search of each file's *contents*. It's pretty fast, even when searching a floppy disk.

Word Search allows any combination of words up to 20 letters, and can link words using logical operators representing AND and OR. Entering "Commodore;Amiga" yields a list of all files containing both "Commodore" and "Amiga." Entering "Commodore,Amiga" yields a list of all files containing either word. Pattern matching is allowed, using standard MS-DOS wildcards of "*" and "?." Entering "s?t" finds the words set, sit and sat. Entering "s*t" locates all files containing set, sit, and sat, as well as sunset, spirit, and sweet.

Multiple Documents

This is one area where Amiga WordPerfect leaves its befuddled MS-DOS cousin in the dust. While the MS-DOS version of WordPerfect allows users to switch between two documents in memory at a given time, Amiga users are not so constrained—in fact, the WordPerfect manual says you may have up to 32 *windows* open at once if system memory allows! (I sincerely hope I never need to put that claim to the test.) WordPerfect also allows easy cut-and-paste between multiple documents.

HOW DO YOU SPELL RELIEF?

WordPerfect's spelling checker features a dictionary of over 115,000 words, and allows you to create customized supplemental dictionary files. If you mistakenly add a misspelled word, you can go back and remove it from the dictionary. In addition, the WordPerfect spelling checker performs word counts.

When you call the spelling checker, you are given several options: to check one word, a page, a predefined block of text, or a whole document; to change dictionaries; to look up a spelling; or to perform a word count.

The WordPerfect spelling checker is smart; unlike some other programs, it is not easily tripped by punctuation and plurals. Besides checking for misspellings, it also detects two occurrences of the same word in a row.

When the spell checker encounters a word not found in its dictionary, it stops, flags the word, and displays a list of the most likely correct words. To select one of these words, simply point and click, or select its letter. If the spelling checker can't find a likely match, you may manually edit the word, query the dictionary phonetically or using pattern matching, or—if the word is correctly spelled—you may add the word to the supplementary dictionary or skip over the word entirely.

One feature I particularly like is WordPerfect's handling of words containing numbers. Upon the first occurrence of a word containing a number, the spelling checker offers you the option of ignoring *all* words containing numbers. This is a big help in technical writing.

One possible drawback to WordPerfect's spelling checker is its speed. On floppy-based systems, the spelling checker takes time—nearly 17 minutes to check a 4,600 word file. In fairness, this is AmigaDOS's rap as much as the program's—AmigaDOS has never been renowned for its floppy disk speed. When tested on a SupraDrive 20 megabyte hard drive, the same file took considerably less time--approximately six minutes. The best speed is achieved by loading the spelling checker's dictionary into the ramdisk; under this configuration, the spelling checker whipped through the file in one minute, twenty seconds. This option is not for everyone—the dictionary takes up over 260K—but even at that size, an expanded Amiga 500 could handle it with ease.

What's The Word?

Perhaps your problem is not spelling, but trying to decide which word to use. WordPerfect's got that covered too, with an impressive thesaurus utility.

Let's say you need another word for "run." To call up WordPerfect's thesaurus, place the cursor on the word "run" and press ALT-F1. Within seconds, a window opens up listing a series of headwords broken into subgroups.

A headword is a word that may be immediately substituted into the text or further explored within the thesaurus. A subgroup is a collection of headwords sharing the same connotation. This is important with a word like "run," which has several meanings. In the first subgroup, WordPerfect's thesaurus lists bolt, dart, dash, race, and sprint. Entirely different meanings for "run" are found in the second subgroup, which includes function and operate. Another subgroup lists administer, govern, and manage; yet another, drive, maneuver, and propel. (There are three more subgroups, but I'll be merciful.)

(continued)

If you see a headword that appeals to you, you can insert it into your document at this time. If one looks close, but isn't exactly what you're looking for, you can break it down further—a double-click on "race," for instance, yields nouns as diverse as competition, chase, and species, and verbs such as hurry and speed, and (my favorite) an antonym, mosey.

By and large, the thesaurus works well. It has a generous vocabulary and locates word lists quickly, even from a floppy disk. One warning: the version I tested, dated September 27, 1987, has a particularly virulent bug infecting its View Doc option. View Doc is supposed to allow you to examine your document for context while in the Thesaurus. However, by repeated testing, I found that clicking the mouse pointer in the document window and pressing the cursor keys a few times caused a full system crash. By the time this article is printed, a fix should be available—registered owners are advised to call WordPerfect's Customer Support hotline for more information.

Hard Copy

WordPerfect handles document printing as a separate "task"—it even has a separate "Print" icon on the main system disk, which may be used apart from the main WordPerfect program.

The biggest advantage to this approach is, once you tell WordPerfect to print a document, you can resume writing without waiting for it to finish; you can return to your document immediately. That's nice. WordPerfect automatically spools multiple print jobs into a queue and allows you to temporarily suspend, cancel, or "rush" jobs within the queue.

WordPerfect allows you to print an entire document, a selected page, or any range of text identified within a block.

WordPerfect takes a radical departure from previous Amiga programs in its printer handling. While Amiga WordPerfect uses the Preferences printer driver definitions if you insist, its full power is realized by selecting one of the 250 custom drivers contained on the Print disk. You may select up to six different printers, configuring each to suit specialized needs: number of copies, print to a DOS text file or other device, binding width offset, continuous-form or cut-sheet paper. WordPerfect also supports sheet-feeders.

In the unlikely event that your printer is not among those listed on the Print disk, the disk contains a PrintDef program, whereby you may define new printer drivers or modify old ones. The Print disk also contains a 30-page WordPerfect-format document explaining use of the PrintDef program in detail. It's a rather strange Catch-22 that this document is not provided in hard copy form, considering a good portion of those who will need it the most are those having trouble getting WordPerfect to work with their printers!

Most people will not have problems with WordPerfect's printer drivers. One phenomena—printers offering multiple emulations—has given the customer support specialists a real workout. The Star SG-10, for instance, supports either IBM printer codes or its own quasi-Epson "Star" mode. None of the drivers included with WordPerfect adequately support "Star" mode—the printer's out-of-the-box default! Star SG-10 owners are advised to flip their printers' dip switches to the IBM position.

Unlike the MS-DOS WordPerfect, the Amiga version does not support the PostScript printer language that has become popular with many laser printers. Postscript support is planned for a future update.

Advanced Features

If WordPerfect offered only the functions described so far, it would still be a strong contender—but those are just the warm-up. Advanced features help WordPerfect really outdistance the pack. There isn't room to do much more than touch lightly in each of the advanced areas, but even that should give you an idea of the program's capability.

Tables Of Contents And Indices

If you have ever been through the painful experience of manually assembling a table of contents or an index (extra credit if you did it on a typewriter), you will probably walk away muttering the first time you see how easy WordPerfect makes these two most hated and feared word processing tasks. The process is nearly identical for tables of contents and indices. In addition, WordPerfect allows you to define up to five lists which may be used to show directories of figures or graphs.

The first step in creating a table of contents, index, or list involves marking each occurrence of the word or phrase you want to include. This may be done as the document is written, or on the final draft. Automating the process with a well-constructed macro saves a lot of time. Tables of contents and indices may be designed on multiple levels; for instance, under the contents subheading "GOVERNMENT REGULATIONS," you could include separate subheadings for Safety, Health, and so forth.

Next, you must tell WordPerfect where to put the table of contents, index or list, and which of several formatting options you want to use.

The last step is the easiest: GENERATE. When you select the Generate option from the Mark Text menu, WordPerfect scans the document for marked words, sorts as necessary, formats the tables, and inserts them in the document. It's pretty impressive—the first time I saw the process I could only shake my head in disbelief.

If the changes affect page numbering or table headings, selecting Generate again reformats a fresh, updated table of contents and index.

Automatic Outlining

WordPerfect's outlining ability is so intuitive, it borders on artificial intelligence. All you need to do is decide at what level each outline item appears by pressing the Tab or Indent keys; WordPerfect numbers each item appropriately—Roman numeral capitals (I., II., III.) for the first level, capital letters (A., B., C.) for the second level, numbers (1., 2., 3.) for the third level and so on. Should you get part of the way through an outline and realize you left out an item half a page back, no problem—WordPerfect renumbers your outline to accommodate the new entry.

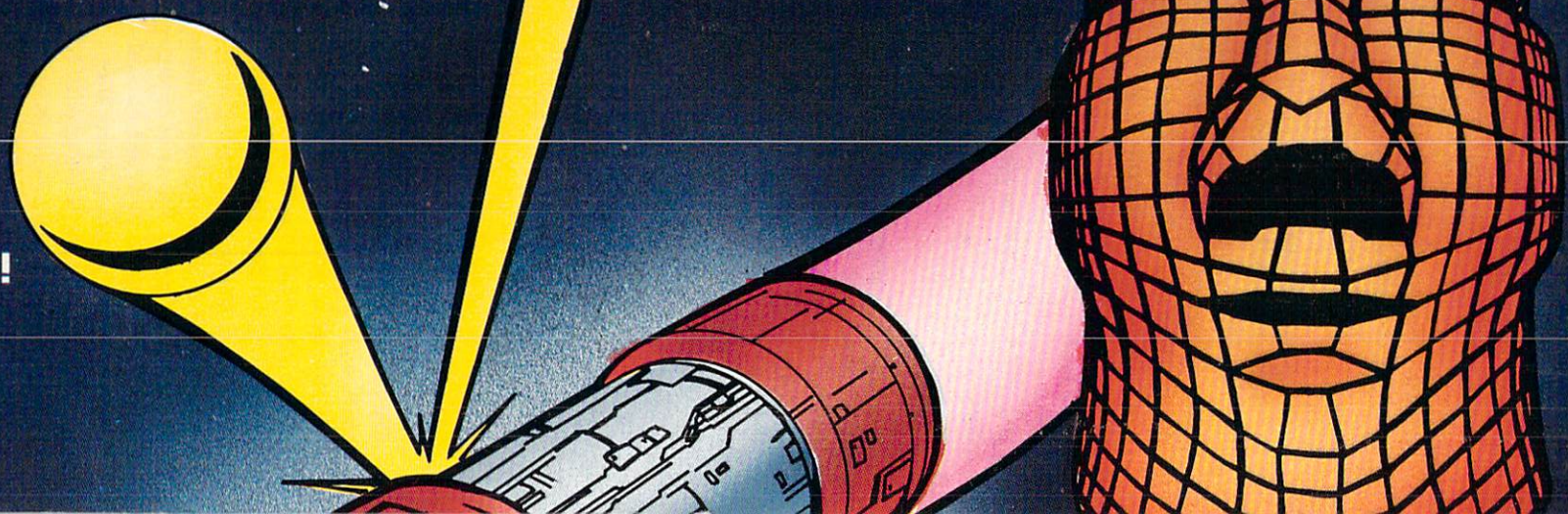
You are not limited to the "standard" outlining format described in the previous paragraph. WordPerfect also outlines in "paragraph" notation (1., a., i.) and legal notation (1.1, 2.1.1). If its built-in outline formats don't suit your needs, WordPerfect allows you to create your own customized notation format for up to seven levels. Best of all, you can change notation on a completed outline from one style to another with just a couple of keystrokes.

(continued on page 66)

ARKANOID™

THE #1 ARCADE BLOCK-BUSTER!

- **33** Sizzling Levels of Play!
- **Dazzling** Graphics, Digital Stereo Sound
- Fast! Fun! Furious! . . . Frustrating!
- A Breakthrough in Arcade Realism
- Arcade High Score Table and Demo Mode
- Never Needs Quarters



ONLY $49.95 * !

SIMPLY THE BEST AMIGA GAME EVER!

MARAUDER

"Marauder II is an excellent utility at a bargain price."
Bob Ryan-Amiga World

"Every Amiga owner should have a copy . . ."
Bruce Webster-BYTE

"If you own an Amiga and don't have Marauder II from Discovery, you don't have one of the best tools an Amiga owner can have." -INFO Magazine

The best copier for Amiga™ at any price! Still only **$39.95** *

NEW! **DX SERIES**

Our new **DX SERIES**™ provides the power and features of the world renowned Hewlett-Packard advanced programmable calculators at a fraction of the price. The **DX-16**™ is a perfect on screen replica of the HP-16C™, the ultimate programmer's calculator. The **DX-11**™ mirrors the HP-11C™, an excellent choice for students, engineers and general purpose calculations. Both are included to make this package a double value. Both are fully programmable and fully multi-tasking power tools. Complete tutorials and excellent documentation will get you started quickly. **$49.95** *

NEW! **AMNIX**

AMNIX™ delivers the finest features of the famous UNIX™ C-Shell. Free yourself from constant disk swapping. Over 40 high-speed resident commands make their disk-based Amiga DOS counterparts obsolete. **AMNIX**™ also provides extended batch files, command line history and editing and environment variables. It's easy to learn and superbly documented. A "must have" for Amiga power users! **$49.95** *

Amiga™ is the trademark of Comodore-Amiga, Inc.. UNIX™ is the trademark of Bell Laboratories Inc.. HP-16C™ and HP-11C™ are trademarks of Hewlett Packard Co.

LIMITED TIME OFFER!

ORDER DIRECT AND GET
$5.00 OFF
EACH PRODUCT YOU BUY
PLUS

FEDERAL EXPRESS

ABSOLUTELY FREE!!

TO ORDER CALL TOLL FREE
1-800-34-AMIGA
or (301) 268-9877

or send check, cash or
VISA or MasterCard numbers to:
Discovery Software International Inc.
163 Conduit Street
Annapolis, MD 21401

*Order now for $5 introductory discount!

(Federal Express shipping in continental U.S. only.)

VISA OFFER SUBJECT TO CHANGE WITHOUT NOTICE MasterCard

(continued from page 63)

Multiple Columns

Multiple column format is another area where WordPerfect simplifies what has typically been an exercise in frustration. You may define up to five columns across a page. Depending on how many columns you select, WordPerfect suggests the individual column margins, or you may supply your own.

Two types of columns are available. "Newspaper" format is commonly used in newsletters where text continues from the bottom of one column to the top of the next. The "parallel" format is used for information that must be aligned *across* columns. The WordPerfect tutorial uses an example of an address list containing names in column 1, associated addresses in column 2, and phone numbers in column 3 to illustrate this format.

The best thing I can say about WordPerfect's handling of multiple columns is that it frees you from concentrating on the how's of formatting, allowing you to concentrate on writing, instead of keeping two sets of margins straight.

The Miracle Of Automation

WordPerfect's Macro abilities offer unprecedented flexibility and power, allowing you to automate frequently used functions. WordPerfect's Macro Define allows you to you easily (yes, easily) create macros to format documents, insert text, save and print documents, and even call other macros. In fact, WordPerfect macros can automate any series of tasks that can be executed sequentially. My personal favorite macro, designed by Rhyder McClure, a columnist for *PC Magazine*, uses WordPerfect's math functions to calculate and print invoices to a laser printer! We're talking serious power here.

We're also talking *accessible* power here. WordPerfect's Macro Define function can be compared to a "keystroke recorder"; once activated, it remembers every keystroke entered (commands as well as text) until you tell it you're through. Running the macro simply plays the keystrokes back.

You may name a macro using any valid filename, though one-letter macro names allow you to execute the macro by simply pressing an Amiga key and the letter—for instance, executing a PRINT macro with Amiga-P. You may define as many macros as you like—WordPerfect saves them as permanent files. Once defined, you may use a macro in subsequent sessions.

Lessons 16 through 20 in the documentation's tutorial section offer step-by-step examples of the power of WordPerfect macros, guiding new users through the creation of both simple and complex macros. The examples are practical as well—by the time you get through the four lessons, you will have created a two-keystroke macro to save and print files, another macro to add a closing salutation to correspondence, and a third macro to further automate the creation of indices.

WordPerfect macros may repeat, chain to other macros, and even branch based on what they encounter at run time. Once a macro has been created, though, there is no way to return to either examine the macro for its function (short of executing it) or edit it. This a drawback. A Macro Editor was such a popular feature of the MS-DOS WordPerfect Library (a collection of add-on utilities), that WordPerfect Corporation recently began marketing it separately. WordPerfect Corporation plans to demonstrate an Amiga version of the WordPerfect Library at this fall's COMDEX—unfortunately, they are not currently planning to include the Macro Editor in the Amiga version. Unlike document files, macro files are not compatible across machine systems.

Some standard Amiga key conventions forced the programmers of Amiga WordPerfect to make a departure from the standard WordPerfect method of using the ALT key to call one-letter macros. Standard Amiga key conventions reserve the ALT key for overstruck characters needed for some foreign alphabets. Considering that approximately half of Amiga WordPerfect's sales to date have been in European markets, leaving that function intact was probably wise.

Fixing one problem created another, though—the Amiga has key conventions for its "Amiga" keys, too! Amiga-N and Amiga-M, for instance, may be used to toggle between screens. The designers fixed this problem by assigning certain Amiga-key combinations as Amiga-keys, and others as keys as WordPerfect macros. I found this out when my copy of WordPerfect stubbornly refused to let me name a macro "Amiga-S." I thought I had a major bug on my hands, but that was not the case.

Keys may be assigned as standard Amiga keys or as macros through Screen (CTRL-F3) option 2, Ctrl/Amiga Keys. To change an Amiga key from "Amiga" to "Macro," click on the window next to the letter you want to change, enter the number "1," and press return. You read it here first—it's not in the manual.

Mail Merge

WordPerfect's mail merge facility allows you to generate hordes of "personalized" form letters just like the big boys. To perform a mail merge, WordPerfect requires you create a primary file, containing the document which will have the information inserted at set points, and a secondary file containing the information to be merged with the document.

Like many database programs, WordPerfect inserts fields in the secondary file into predefined points in the primary file. Unlike most database programs, WordPerfect maintains the document's format through diverse field lengths, and wrapping words when necessary, just as if the document had been typed by hand. In one merge example included in the documentation, one field ranges in length from two lines in one record, to six lines in another.

The finished merge may be sent to the editing screen for review, or directly to the printer. Besides the codes entered from the secondary file, you may also configure WordPerfect to accept keyboard input during the merge process. You may even execute macros during the merge.

Overall Impressions

WordPerfect Corporation did what it set out to do—redefine the standard of Amiga word processing. Love it or hate it, WordPerfect will be the standard against which all others will be measured for years to come.

This compatibility isn't one hundred percent; Amiga WordPerfect does not support sorts, and its "border draw" function is anemic compared to the MS-DOS version's Line Draw. Amiga WordPerfect's requesters are also a bit more sluggish than you might expect from a product written totally in 68000 assembly language. Finally, if you require options specific to WordPerfect 4.2, you should know that the Amiga version is written at level 4.1.

Even with these shortcomings, the Amiga version essentially retains all the power of other machines' versions, while adding extra measures of capability and ease of use. Overall, I think the Amiga version comes out on top.

Is WordPerfect for everyone? Probably not. If you are contemplating a word processor strictly for home use, you will be happy with WordPerfect—though it'll be a little like driving a Maserati to the corner store. And there's that Maserati price tag, too—even with WordPerfect's suggested list price of $395 discounted as heavily as 50% in some areas, most people will have to think twice before making such an investment.

Then again, when you buy WordPerfect, you aren't just investing in the product—you're investing in the *company*. Their customer support is the best in the business: a toll-free 800 number is available Monday through Friday on an unlimited basis to all registered users. If you find a bug, you won't have to wait for a major upgrade—WordPerfect Corporation is ruthless about bugs, and not shy about upgrading software. You can expect a fast replacement, free of charge, with no hassles.

For serious word processing on the Amiga, WordPerfect is the #1 choice. And when compared feature-for-feature against its competition, there is no #2.

•AC•

WordPerfect for the Amiga
Suggested list price: $395
Not Copy-Protected

WordPerfect Corporation
288 West Center Street
Orem, Utah 84057
(801) 225-5000

The Insider and KwikStart
from Michigan Software

by Ernest P. Viveiros, Sr.
Hardware Editor

The Insider™ is a state-of-the-art RAM expansion board for the Amiga. The unit is mounted inside the main Amiga™ console—hence the name "Insider." There are both pros and cons to any interfacing scheme, and every system designer is certain of only one fact: You can never please everyone.

There are only three options available when it comes to expanding the available RAM in the Amiga 1000. You can expand the original 256K system to 512K using the Commodore™ 256K RAM expansion unit or a similar expansion unit from an independent developer. This stage of expansion is relatively easy to accomplish since RAM expansion boards simply plug into the RAM expansion port on the front of the Amiga system console—a simple, five-minute installation.

However, this is the maximum memory (256K) that can be added by the expansion port. 512K is very respectable by early 1980s standards, but, with today's relatively low-cost RAM chips and the sophistication of existing memory-intensive software, it simply does not cut the icing on the cake—not for me anyhow.

Your second option involves the expansion bus located on the right side of the Amiga console. Although Commodore does not currently use the Amiga expansion bus, several independents offer RAM expansion units which can take the Amiga to its current maximum address space of 8 megabytes. There is only one disadvantage that comes along with these super megabyte RAM expansion units. Simply put, they are substantially more expensive than the Insider. The difference in cost is justifiable, however, simply because a 2, 4, or 8 megabyte RAM expansion is vastly more complicated (and costly) to produce.

Your final option is to hack the RAM expansion yourself. A simple 512K RAM expansion was presented in *Amazing Computing V2.1.*

Michigan Software has put together a combination of hardware and software sure to please most Amiga users. The specifications of the Insider RAM expansion board are as follows:

1. One megabyte of expansion RAM.
2. No forced wait states.
3. Transparent refresh.
4. Only 600 ma draw from power supply.
5. Real-time clock with 10-year lithium battery back-up.
6. Auto-configuration under 1.2.
7. Software which includes:
   a. Addmem program.
   b. Memory test program.
   c. Real-time clock program.
   d. Ram on/off program.

A comprehensive installation manual and a one-year warranty are also included with the Insider RAM expansion unit. The warranty does not include damage incurred during installation. The Commodore warranty does not cover any damage you cause either. You're on your own, but, if you follow the instructions closely, little can go wrong.

The Insider does not fall into any of the original RAM expansion options described earlier. It falls somewhere between the hack and expansion bus options. I use the term hack simply because the Amiga must be disassembled to install the Insider. The Insider is a professionally designed, high quality product that uses state-of-the-art components and technology. This advanced technology, including the LSI RAM controller, clock module, and PALs, allows the Insider to exist in its current configuration and fit easily into an area where no system expansion was planned.

The Insider is well thought-out and designed for installation by anyone, even those without electronic knowledge. Three screw drivers are required—a Phillips head, a small flat blade, and a medium flat blade. A pair of long needle nose pliers will also save you a lot of aggravation.

The installation manual takes you step-by-step through the entire installation procedure. The manual is excellent; nothing seems to be missing. The folks at Michigan Software have even provided a small box at the beginning of each instruction, so you can check off the steps as you complete them. This may seem trivial, but it shows that much thought went into easy installation.

The following is a general description of how to install the Insider. (You should know the steps involved before you spend your hard-earned money on a project you may not want to complete.)

1. Remove all cables that are externally attached to the Amiga.

2. Remove 256K RAM expansion card (if used).

3. Remove all hardware holding the Amiga case closed.

4. Open the Amiga case.

5. Remove all hardware and bend all tabs securing the RF shield to the Amiga mother board. Remove shield.

6. Remove mounting hardware for the disk drive and remove the cables attached to the drive. Remove the drive from the Amiga. (The Insider can be installed without removing the disk drive unit; it just takes a little more time and patience.)

7. The MC68000 microprocessor must now be removed from its socket. Be very careful here because you're dead in the water if you break it. It doesn't cost much, but you can't get one at Radio Shack™.

8. Plug the MC68000 microprocessor into the socket provided on the Insider expansion board.

9. Plug the provided MC68000 extension socket into the Amiga mother board.

10. Plug the Insider into the extension socket.

11. Attach four jumper wires to the Amiga using test clips. (No soldering is required, but you can solder if you have the required expertise.)

CAUTION: It takes considerable force to insert the MC68000 extension socket into the Amiga microprocessor socket. This pressure must be exerted again to insert the Insider into the extension socket. Commodore did not design the mechanical supports for the mother board for any internal system expansion, so there are no support posts beneath the microprocessor socket. Excessive flexing of the printed wiring board can break the circuitry.

Well, that's all there is to the installation. It seems complicated, but it's a piece of cake, and the results are just as sweet. Before you close your system, test the RAM expansion—after all, we've all heard of Murphy's law.

The exact steps for testing the system depend on the version of Kickstart™ you use. Kickstart version 1.2 causes the Insider to auto-configure if the default switch settings are used. The Insider has a four-position dip switch used to set the address range of the expansion RAM. The corresponding address ranges for the possible switch settings are:

Although the Insider RAM can be installed at any usable address range, it will most often be installed at the auto-configuration default setting of c00000 cFFFFF (since most users are currently running the latest version of Kickstart).

If the Insider is configured at any address other than c00000 cFFFFF, or if the Amiga is running version 1.1 of Kickstart, you must use the ADDMEM utility to inform the system of the added memory. Run the program from the CLI each time you boot the Amiga or modify the start-up sequence. This way it's done automatically.

Dip switch settings				Range (notes)
1	2	3	4	
off	off	off	off	do not use
on	off	off	off	do not use
off	on	off	off	200000 2FFFFF
on	on	off	off	300000 3FFFFF
off	off	on	off	400000 4FFFFF
on	off	on	off	500000 5FFFFF
off	on	on	off	600000 6FFFFF
on	on	on	off	700000 7FFFFF
off	off	off	on	800000 8FFFFF
on	off	off	on	900000 9FFFFF
off	on	off	on	do not use
on	on	off	on	do not use
off	off	on	on	c00000 cFFFFF **
on	off	on	on	do not use
off	on	on	on	do not use
on	on	on	on	do not use

Several other utility programs are provided on the disk. Insider's real-time clock is running when you receive it, but there must be a way of setting the clock and transferring the time and date information to the Amiga system software. This function is provided by the RTclock program which can be run from the CLI, or, preferably, by modifying the system start-up sequence. The program provides a basic instruction set which can both set and query the Insider clock. Once the clock is set, you'll never have to type the time or date again.

The software also includes a memory test program, Memtest, and the RAM on\off software which allows the user to turn the extra RAM on\off from the Workbench. Not all software is created equal. Many early programs were not designed to run outside the Amiga's original 512K of RAM. If you have such a program, a simple click on the RAM on\off icon in the Workbench™ allows you to run the program without problem.

With the Insider, 256K systems increase system RAM to approximately 1.2 meg., and 512K systems peak at 1.44 meg. These figures will vary somewhat, according to variations in start-up sequences.

(continued)

I have had my Insider for over a month, and I have not experienced any hardware or software problems. Is it worth it? In my opinion, it's worth every penny.

KWIKSTART

I had just closed my Amiga after installing the Insider RAM expansion board when, lo and behold, the UPS™ truck delivered Michigan Software's new, exciting Kwikstart.

Kwikstart is a ROM add-on board for the Amiga 1000. Since some readers may not be familiar with the term ROM, a brief explanation is in order.

ROM is read only memory. ROM, just like RAM (random access memory), can be programmed. The big difference is, once programmed, the information stored on ROM cannot be changed by the user. Additionally, unlike RAM, ROM does not require any power to maintain the stored information.

Information stored on ROM also executes (runs) at a much higher speed, because ROM memory is directly connected to the microprocessor bus. Intervention is not required by hardware (disk access) or software conversion routines. ROM memory appears to be the ideal form of information storage with one possible exception: It is permanent, and no alterations are possible.

When Commodore released the 1000 and its system operating software, certain bugs existed; other problems were uncovered during initial use. Therefore, they decided against committing the operating system software to ROM upon the release of the Amiga 1000. As with everything else, change is inevitable. Version 1.2 of the system software has been released by Commodore as the *de facto* standard.

The ROM installed on the Kwikstart expansion board contains version 1.2 of the Commodore Amiga system operating software.

Installation of Kwikstart is similar to installation of Insider (although Kwikstart does require more hands-on ability). The main difference is the removal of one component (PAL) from the Amiga Kickstart board. Otherwise, the installations are nearly identical. The steps required to install the Kwikstart ROM board into the Amiga main console are:

1. Remove all cables connected to the Amiga's main console.

2. Remove the 256K RAM expansion board installed at the front of the Amiga main console (if used).

3. Remove all hardware holding the Amiga case together.

4. Remove the RF shield.

5. Remove the daughter (Kickstart) board from the Amiga mother board.

6. Remove (unsolder) the PAL at location 6J on the Amiga daughter board.

7. Solder the provided socket into place at location 6J on the daughter board.

8. Assemble the new PAL into the socket on the daughter board.

9. Re-install the daughter board onto the Amiga mother board.

10. Remove the disk drive. *(Note: The Kwikstart board can be installed without removing the disk drive; it just takes more time.)*

11. Remove the MC68000 microprocessor from its socket.

12. Install the MC68000 microprocessor into the socket on the Kwikstart board.

13. Press the Kwikstart board into the microprocessor socket.

14. Re-mount the disk drive.

15. Solder the two wires attached to the Kwikstart board to the Amiga mother board.

Before closing up the Amiga, test the installation.

Kwikstart is shipped from the factory configured to power-up under the Kickstart™ 1.2 ROM version. In that case, you are almost immediately greeted by the INSERT WORKBENCH screen. If you want the Amiga to power up normally under the disk-based version of Kickstart, you must switch a jumper on the Kwikstart board. The jumper is a simple press-on type device—no soldering required.

Why would anyone want the Amiga to request the disk-based version? The answer is complicated, but simple.

When a system manufacturer comes out with a new product, he issues guidelines to hardware and software developers. These guidelines provide a path for both hardware and software to be upwardly compatible with future versions of the system software. Compatibility becomes an issue because these guidelines may not always be followed.

Many Amiga software packages do not run, or are hampered by bugs when run under Kickstart 1.2. Most current Amiga users are likely to own at least one of these packages, so Michigan Software had the forethought to offer the ROM/DISK Kickstart option. The goodies do not end here.

To make the Kwikstart board as flexible as possible, Michigan Software also offers a way to switch, mid-stream, between the disk-based and ROM-based Kickstart. A brief example is the simplest way to explain this option. Let's assume the Kwikstart ROM board is configured so the Amiga powers up under the ROM version of Kickstart.

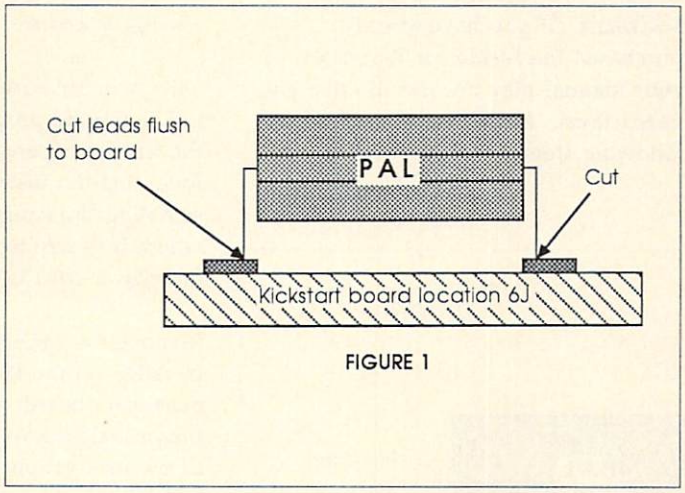


FIGURE 1

Approximately two seconds after the power has been turned on, you see the INSERT WORKBENCH screen; that's pretty fast when compared to the disk-based version. If you reboot the system by pressing the Control Amiga/Amiga keys for less than three seconds, you again see the INSERT WORKBENCH screen. However, if you do a system reboot by pressing Control Amiga/Amiga for more than three seconds, you see the Insert Kickstart greeting. This same sequence of events also holds true for a Kwikstart board configured to bring the Amiga up under the normal Kickstart environment. In effect, you can toggle between the ROM and disk-based Kickstart.

To install Kwikstart, you must remove a PAL device from a printed wiring board and solder a socket onto the board. Removing the old PAL is quite difficult, but soldering the socket is relatively easy. Be careful, though. Removing anything from a printed wiring board involves a certain amount of risk, even for someone with experience. Michigan Software's concern is evident in the installation manual—several pages are devoted to removing the required PAL device.

It took me ten minutes to unsolder the PAL and solder in the socket, but I have pretty good equipment and more years of experience than I care to admit. If you are a complete novice, I have only four words of advice for you: Do not do it. Instead, get help from a friend who has experience removing components from printed wiring boards or pay a professional to do it for you. I'm not trying to scare you off; it's really a simple job, but why take a chance? The Kwikstart board is worth what you may have to pay to have the PAL removed.

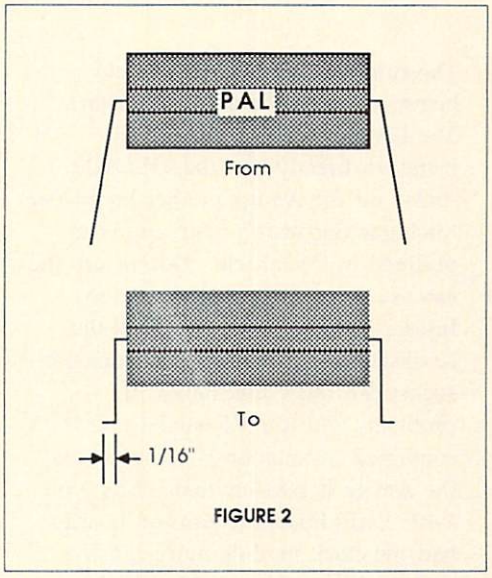


FIGURE 2

You have one other option available, although I am somewhat hesitant to recommend it because it will probably void Michigan Software's warranty.

Using a fine-pointed pair of diagonal wire cutters, cut through all leads of the PAL at location 6J on the Kwikstart board. Trim the leads as close as possible to the Kickstart board, but be careful not to cut into the traces or pads on the Kickstart board (see Figure 1).

Next, prepare the new PAL for mounting, using the following alternate method. The leads on all integrated circuits are shipped from the manufacturer bent at a slight angle from the body of the IC. These leads must be made parallel to the IC's body prior to installation onto the board (see Figure 2). That relatively easy task can be accomplished by using either the instructions provided in the installation manual or by individually forming each lead. Next, the leads must be formed at a 90-degree angle to the main IC body (see Figure 3). The leads must also be trimmed to the approximate dimensions shown.

This alternate assembly method requires that the leads on the new PAL and the termination points to which it is to be soldered both be pre-tinned prior to soldering. Tinning is simply the application of a fine coating of solder to a terminating lead or wire. When you tin the pads at location 6J, try to produce solder joints which are smooth, flat, and approximately the same height above the printed wiring board surface.

The new PAL can now be directly soldered into place. Orient the PAL to the proper position and place it directly onto the solder pads at location 6J. Visually verify that all the component leads are making full contact with their terminating points.

(continued)

You may have to reform some of the leads to get proper alignment. Once you are sure of the alignment, solder the four corner leads of the PAL to the board to hold the device in position. If you have tinned both the PAL and the board properly, no additional solder is required; simply reheat the existing solder so it melts and flows together. Solder the remaining leads to the board.

What do you get for your money by installing Kwikstart?

1. Increased system performance—an almost instantaneous Workbench Screen.

2. System flexibility—the disk-based versions of Kickstart can be called up. This could be an invaluable commodity if a higher revision Kickstart or other system operating software becomes available in the future.

3. Approximately 256K of additional memory is available when the ROM based Kickstart is used.

4. The satisfaction of knowing that you have not spent your hard-earned dollars fruitlessly on an antiquated system (The Amiga 500 and 2000 both have Kickstart installed in ROM.).

Dual Installation

The Insider RAM expansion board and Kwikstart ROM board by Michigan Software are completely compatible and can be simultaneously installed into the Amiga system console.

The combined installation is approved by Michigan Software. In fact, it was originally part of the design concept for the two units. Michigan Software has added the necessary instructions for dual installation to the manuals shipped with current Insider and Kwikstart. If you have already purchased the Insider or Kwikstart, your manual may not include the new instructions. In that case, note the following steps:

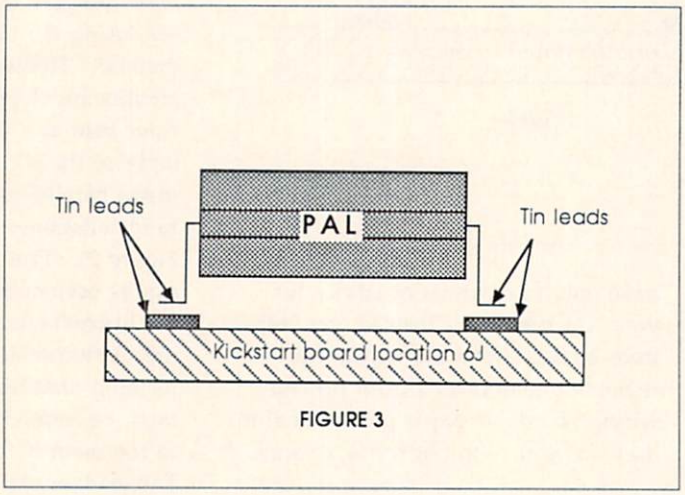


FIGURE 3

The combined installation should begin with installation of Kwikstart. The Kwikstart board should be installed directly into the MC68000 socket on the Amiga mother board per Michigan Software's instructions as outlined in this article. Do not use the extension socket provided with the Insider installation kit to install the Kwikstart ROM board. The extension socket can cause mechanical fit problems with the RF shield after the combined installation is complete and the Amiga is reassembled. Why, you ask? Early Insider expansion boards had the clock module mounted in a socket. When this extra height is added, only a marginal fit exists (by my calculations anyway—since my Insider has the clock module mounted directly onto the board, I cannot be sure).

After installing the Kwikstart ROM board, test it thoroughly before installing the Insider. Unfortunately, testing Kwikstart requires partial reassembly of the Amiga, but it's essential for troubleshooting the installation. The installation manual describes the testing procedure.

Once you are sure the Kwikstart board is functioning properly, you can install the Insider expansion board. Begin by following the instructions for disassembling the Amiga to the point where it is safe to work on—be sure the power cord is removed.

Remove the MC68000 microprocessor installed on the Kwikstart ROM expansion board; again, follow all prescribed precautions. The remainder of the installation is relatively easy. Press the Insider into the socket on the Kwikstart ROM board by alternately applying pressure to each end of the microprocessor socket mounted on the Insider until the pins are fully seated into the Kwikstart board. If you have not already done so, mount the microprocessor into the socket on the Insider and attach the required wires to both the Kickstart and Amiga mother boards. Don't forget to install the jumper which ties together the Kickstart and the mother boards.

The installation is complete! Test the combined expansion boards before closing up the Amiga ... and wipe the perspiration from your brow.

The old adage—you can't have your cake and eat it too—doesn't apply when it comes the Insider and Kwikstart expansion boards. You don't have to choose between them—you can have both.

•AC•

Forth!

by Jon Bryan

Translate the RKM DumpRPort example from C into Multi-Forth

This month, I have several useful additions to your Multi-Forth toolbox. A member of the Forth forum on CompuServe was having difficulty with the DumpRPort example that begins on page 423 of the *ROM Kernel Manual: Libraries and Devices*. This problem struck me as an excellent exercise, so I translated the C source into Multi-Forth.

Aside from providing a good example of a straight C-to-Forth translation, the program also includes the definition of the IntuitionBase structure and examples of the procedures for opening libraries and using the printer device. DumpRPort is essentially equivalent to the GraphicDump utility in the Workbench System drawer. After a delay of a few seconds to allow you to arrange the display to your liking, DumpRPort dumps the screen to the printer.

The first thing you'll see in the example is the phrase "SYMTABLE DEFINITIONS." Creative Solutions' conventionally places structure definitions (symbols) in the Symtable vocabulary. If you are unfamiliar with the concept, a vocabulary in Forth is a mechanism for structuring the dictionary similarly to the subdirectories on a floppy or hard disk. The vocab mechanism allows words with the same spellings to have different meanings, depending on the context. It also subdivides the words in the dictionary into groups according to their function, as this case indicates. SYMTABLE DEFINITIONS links new words into the Symtable vocabulary.

The data structures needed to dump the screen are then defined. Structures located in other files are compiled if they aren't yet in the dictionary. Each C include file has a counterpart in Multi-Forth (translated by yours truly). Each file begins with a null definition which is used as a flag. For instance, the word EXEC_TYPES_F indicates that the types.f file in the Exec subdirectory has already been compiled. After the exec/types.f, intuition/intuition.f, and devices/printer.f files are compiled, we arrive at the definition of the IntuitionBase structure.

IntuitionBase provides access to certain information which is not available anywhere else. In this case, it is used to retrieve the address of the frontmost screen. It also contains pointers to the currently active Window and Screen, the X and Y coordinates of the mouse, and other pieces of information which can be extremely useful.

The IntuitionBase structure defined in this example program is actually a subset of the complete IntuitionBase area. If you want to extend the structure, you are directed to the intuitionbase.h file in the C includes. Be warned, however, that the structure will change from one release of the operating system to the next. This structure contains much potentially useful information (including the pointer to the Preferences structure), but the information may not be in the same location when AmigaDOS 1.3 is released. Polite ways to use or change those sorts of things are available to make your operating system much happier.

The "printerIO" structure is a multiform (called a union in C) which is as large as its largest member. After it is defined, you'll notice that we switch back to the Forth vocabulary and make DECIMAL our current base. You will also note definitions for the Exec calls OpenLibrary and CloseLibrary, and translations of the C procedures "CreateExtIO" and "DeleteExtIO."

The source for the C functions is included in Appendix B of *RKM: Exec*. A number of other C "macros" are included in the Exec manual might be worth translating. You should read through at least Appendix B to familiarize yourself with these others.

Three string constants for the calls are available to open the intuition library, printer port, and printer device. For instance, the "OpenLibrary" call gets the pointer to the base of Intuition. The zero-terminated string "intuition.library" then must be passed to the pointer. The string is compiled into the dictionary with the required null at the end using

(continued)

the Forth word "0." This action parses the string (delimited by a closing quote symbol), lays it down in memory, and returns a pointer to the first character in the string. That pointer has simply been turned into a constant.

The latest revision of Multi-Forth includes several enhancements and bug fixes. Notable additions include the "global" variables. Globals have quite a bit in common with "local" variables. For example, they are self-fetching, values are stored in them using the word TO, and ADDR.OF is used to get a pointer to them. The hybrid behavior of the globals puts them somewhere between standard Forth variables and constants. That behavior can come in handy at times, as in this example, where I use the globals to hold the pointers to IntuitionBase, the printer message port, and the printer IORequest. The behavior is appropriate with this type of pointer, since, once established, they won't change. You might think of them as run-time constants.

The definition of DumpRPort is unusual (for Forth) because it expects eleven items on the stack. Don't misunderstand. There's nothing wrong with passing so many parameters on the stack; it's more a matter personal style. I usually try to avoid passing more than three or four parameters to a word because it is much less confusing. The stack manipulations for more items can become very convoluted. Since we're doing a translation of a C program, though, we're stuck with someone else's decisions.

Translating the C version of DumpRPort to Forth using the same order for the arguments presents a problem. We must use the IORequest pointer, and it's deeply buried under ten other items. I hated the idea of doing an "11 ROLL" to get the IORequest pointer up to the top of the stack where it could be manipulated, so I changed the definition to expect the IORequest pointer on top. This alteration seemed appropriate in this case. If we preserve the temporal order of the C example, the IORequest is the last argument retrieved before the call is made to DumpRPort. If you prefer to keep the order consistent with the C function, you'll also want to change the order of "SetupDRP" and "OpenPrinter" in the DumpScreen word.

"GetIntuiBase" obtains the pointer to IntuitionBase. The OpenLibrary call returns the address of the Intuition library if possible. If the library can't be opened, an error is generated. Once the library is open, the IntuitionBase structure provides the offsets to the necessary information. The IntuitionBase structure is never actually used to create a data structure. Its only real function is to provide the offsets into the Intuition library after the library has been opened, since the IntuitionBase structure already exists. It is interesting to note that calls to Intuition functions jump through 68000 jump instructions, located at negative offsets from IntuitionBase.

Ten of the eleven parameters expected by DumpRPort are placed on the stack by the word SetupDRP. The address of the Screen structure, associated with the screen currently in front, obtains the address of that screen's ViewPort. The two pointers then get all the parameters needed by DumpRPort and place them on the stack.

"CleanUp" returns the "printerRequest" and "printerPort" memory to the system and closes the Intuition library once we have finished. Well-behaved applications should not drain any free memory from the system (a rule often violated by early Amiga software).

The printer device must be opened before the screen dump can be sent to it. Likewise, the device must be closed after the dump is complete. In the definition for "OpenPrinter," you may notice that the address of the printer message port is saved in the global variable printerPort, and the printer I/O request pointer is saved in the global printerRequest. If OpenDevice is unsuccessful, the screen cannot be dumped and an error is generated. The error informs the user that the attempt to open the printer device failed.

CleanUp returns the unused printerRequest and printerPort structures to the heap and closes the Intuition library. ClosePrinter simply passes the printer I/O request pointer to the CloseDevice system function.

Finally, "DumpScreen" puts everything together. A delay of six seconds (300*20 milliseconds) gives you time to set up the screen. Next, the Intuition library is opened. SetupDRP places its ten parameters on the stack, the printer is opened, and DumpRPort sends the screen to the printer. When the dump is complete, the printer is closed and the memory which was temporarily allocated is returned to the system. That's all there is to it.

DumpRPort accomplishes precisely the same thing that the C example does, but (as you will see, if you compare the two) in Forth fashion. I think that the Forth version is easier to read than the C example ... but I'm prejudiced. Happy Forthing!

Next Installment

Do you want well-behaved sprites? My latest revision of the sprite compiling tools, along with a change in Multi-Forth, corrects a few of the previous version's problems. Hardware sprites now work just as they should on a machine with more than 512K. The earlier version places sprites in FAST memory where they can't be displayed (I had a 512K machine when I created the tools.). The latest enhancement works together with a change in Multi-Forth to automatically place the images in CHIP memory where they belong, and return the memory to the system when your application exits.

```
\ DumpScreen
\ This is a translation of the RKM:Libraries and Devices
\ example program on pages 424-427.
\ IntuitionBase is an extremely handy structure.
\ DumpRPort should be useful, and a few other goodies,
\ like CreateExtIO and DeleteExtIO, are also included.

\ Since the RKM example is freely distributable, so is this.
\ I'll take credit for the translation, though.
\ J. Bryan:8-9-87

SYMTABLE DEFINITIONS

find EXEC_TYPES_F not
IFTRUE   include exec/types.f   IFEND
find INTUITION_INTUITION_F not
IFTRUE   include intuition/intuition.f   IFEND
find DEVICES_PRINTER_F not
IFTRUE   include devices/printer.f   IFEND

\ A word of warning: There is MUCH more to IntuitionBase than
\ the members presented here.  Almost all of it is forbidden:
\ if you want to know why, read the Rom Kernel Manual.
\ These are the only members "guaranteed" to stay the same across
\ revisions of the Amiga operating system.

structure IntuitionBase
    Library struct: +ibLibNode
       View struct: +ibViewLord
            ptr: +ibActiveWindow
           ptr: +ibActiveScreen
              ptr: +ibFirstScreen
             long: +ibFlags
            short: +ibMouseY
            short: +ibMouseX
             long: +ibSeconds
             long: +ibMicros
structure.end

structure printerIO
    one.of:    IOStdRequest   struct:   +ios
     or.of:    IODRPReq        struct:   +iodrp
     or.of:    IOPrtCmdReq     struct:   +iopc     ;go.on
structure.end

FORTH DEFINITIONS
DECIMAL

: OpenLibrary     ( libName\version - error )     !d0 !a1 exec@ 92 ;

: CloseLibrary     ( library - )     !a1 exec 69 ;

\ The C source for CreateExtIO and DeleteExtIO can be found
\ in RKM:Exec B-9.  The translations are straightforward.

: CreateExtIO ( ioReplyPort\size - IORequest if successful or 0 )
   LOCALS| size ioReplyPort |
  ioReplyPort
   IF   size   MEMF_CLEAR MEMF_PUBLIC or   AllocMem
      dup IF    ( leaves the dup'ed zero on the stack if no memory
)
       NT_MESSAGE   over +ioMessage +mnNode +lnType   c!
        size          over +ioMessage +mnLength          w!
       ioReplyPort over +ioMessage +mnReplyPort       !
    THEN
   ELSE   0    ( if ioReplyPort is null )
  THEN ;

: DeleteExtIO     ( ioExt - )
    ?dup IF   \ "just in case the user did not check things
properly..."
     255 over +ioMessage +mnNode +lnType c!
      -1   over +ioDevice                          !
      -1   over +ioUnit                            !
     dup +ioMessage +mnLength w@ FreeMem
  THEN ;

0" intuition.library" constant intuition.library
0" my.print.port"       constant my.print.port
0" printer.device"      constant printer.device

\ These could just as well be normal variables, but globals
\ work well here so I threw them in as an example.

global IntuiBase
global printerPort
global printerRequest

\ DumpRPort is changed from RKM and expects "request" on top.
\ I hate having to reference the 11th item on the stack!

: DumpRPort (rastPort\colorMap\modes\sx\sy\sw\sh\dc\dr\s\request -
error)
    dup +iodrp   LOCALS| req+iodrp request |
  PRD_DUMPRPORT
    req+iodrp +ioCommand     w!
    req+iodrp +iodSpecial    w!
    req+iodrp +iodDestRows    !
    req+iodrp +iodDestCols    !
   req+iodrp +iodSrcHeight w!
   req+iodrp +iodSrcWidth  w!
    req+iodrp +iodSrcY        w!
    req+iodrp +iodSrcX        w!
    req+iodrp +iodModes        !
    req+iodrp +iodColorMap    !
    req+iodrp +iodRastPort    !
   request DoIO ;
: GetIntuiBase     ( - )
   intuition.library 0 OpenLibrary dup to IntuiBase
   0= error" Intuition won't open!" ;
: SetupDRP     ( - rastPort\colorMap\modes\sx\sy\sw\sh\dc\dr\s )
   IntuiBase +ibFirstScreen @
   dup +scViewPort
   LOCALS| vp screen |
    screen +scRastPort                \ get screen's RastPort
    vp       +vpColorMap   @          \ pointer to ColorMap
    vp       +vpModes      w@          \ modes variable
    0 0                          \ x and y offsets into rastport
    screen +scWidth      w@            \ source size
    screen +scHeight     w@             \ source size
    0 0                           \ dest size 0 because of Special
   SPECIAL_FULLCOLS SPECIAL_ASPECT or ;\print maxwidth w/prop height

: CleanUp     ( - )
   \ The RKM example passes a size to DeleteExtIO (a bug).
   \ Left over, I believe, from a much earlier version.
  printerRequest DeleteExtIO
   printerPort     DeletePort
   IntuiBase        CloseLibrary ;
: ClosePrinter     ( - )     printerRequest CloseDevice ;
: OpenPrinter     ( - request )
    my.print.port 0 CreatePort ( - port )       dup to printerPort
    printerIO CreateExtIO         ( - request ) dup to printerRequest
    printer.device over 0 swap 0 OpenDevice (- request\0=successful )
    IF cleanup abort" Couldn't open printer!" THEN ;
: DumpScreen     ( - )
   300 Delay
   GetIntuiBase
   SetupDRP
   OpenPrinter
   DumpRPort drop
   ClosePrinter
   CleanUp ;
```

•AC•

Modula-2 Programming

on the Amiga™

Part I

CALC: Command Line Calculator

by Steve Faiwiszewski

In the following few articles, I will present CALC, a calculator utility, and some interesting derivations of it. CALC has 26 memory variables and various mathematical functions. All this fits into about 24K of executable code; not bad in comparison to the 28K, four-function calculator supplied with the Workbench. Keyboard use alone seems quite sufficient for this simple tool, so no gadgets, menus, or graphics are used here.

The concept for this tool was originated by Richie Bielak and implemented as a four-function integer calculator a few years ago in Pascal running on a PDP-11. In addition to enhancing the original program to work with real numbers, I've also introduced other capabilities, and ported the utility to TDI Modula-2 and the Amiga.

How to Use CALC

Using CALC is quite simple; you can run it from the Workbench or the CLI. From the CLI, you can open the CALC window by typing "CALC" on the command line. With the CALC window open, you can enter many calculations and use the 26 temporary storage variables. To do a calculation without opening the CALC window, type "CALC" and the expression to be evaluated. CALC prints the result in the CLI window and terminates. For example, typing "CALC 3*15" causes CALC to print "45" and return to CLI.

You enter an expression to be calculated just as you would write it. For example, "CALC5> SIN(LOG(50)*3)+SQR(345)/3" is a valid expression. (In case you're curious, the answer is 6.28023.)

The 5 in CALC's prompt requires that a five-decimal answer be displayed. You can change the precision of the output by specifying "/Dn," where n is the number of decimals to be displayed (1 through 7).

You can assign values to the variables by entering the variable name (A through Z), an equal sign, and the expression. For example, "CALC5> d=5^3" sets the variable d equal to 5 cubed. To examine the value of a variable, simply enter its name. You can even use variables which have been assigned values previously. For example, after you type "d=5^3" and "d*2," d equals 250.

Enter "?" to display a help screen. Enter "/E" to exit the program.

The Program

The program is written in Modula-2 and is made up of four main modules; each module contains procedures that (more or less) logically belong to the same group. The main modules are described below.

The WBStart module determines whether the program was launched from CLI or the Workbench. Before the main routine in CALC runs, the initialization code of WBStart executes. This code first obtains a pointer to its process record and examines the prCLI field. If this field is null, then the program was launched from the Workbench; otherwise, the program was run from CLI. When the Workbench launches a program, it sends it a "Workbench Startup Message," which the program must receive before it does anything else. So if CALC was launched from the Workbench, the WBStart code waits until it receives the Workbench Startup Message. The WBStart module also provides a procedure which replies to the Workbench, to be done just before CALC exits.

The Interpreter module is the heart of the calculator. CALC evaluates the input after analyzing, checking, and converting the expression to some internal representation (which makes it easier to evaluate). More about this module a bit later.

The Interpreter0 module contains miscellaneous routines which could not fit into the Interpreter module, and grew too large to be compiled with version 2.00 of the TDI compiler.

The CALC module (main module) first decides whether to open its own window or to get the input from the command line. If the program was run from the Workbench or from CLI without command line input, CALC opens its own window. This is done very simply with a call to OpenInputOutput. All input is obtained from, and output sent to, this new window. At this point, the module goes into a loop, reading input from the user and processing it appropriately until the exit command is given.

There are two other modules, WriteReal and ReadString, which contain substitutes for certain procedures in the support module library. I didn't like the procedures provided by TDI, so I rolled my own.

The Interpreter

The mechanism for evaluating expression consists of three logical parts:

1. Lexical Analysis (also known as scanning). The expression is broken down to its building blocks, called tokens. For example, the expression "3*(2+4)" consists of the following tokens: 3, multiply-operator, open-paren, 2, add-operator, 4, close-paren. The procedure called nextoken returns successive tokens.

2. Syntax Analysis (also known as parsing). An expression in its original form is not easily evaluated by the computer. Therefore, it must first be analyzed (as well as checked for correctness) and converted into a format the computer can digest. The internal format used here is a binary tree, where each node is an operator that operates on its two leaves. For example, "2+3" is converted to:

```
  +
 / \
2   3
```

Each leaf can also be an operator with two leaves to operate on. The previous example "3*(2+4)" is stored as:

```
    *
   / \
  3   +
     / \
    2   4
```

Functions such as SIN, COS, and SQRT are special; they take only one operand, so they have only one leaf. COS(45) is stored as:

```
  cos
   /
  45
```

The procedure which builds the binary tree (GrowTree) knows operator precedence (e.g., multiplication comes before addition) and builds the tree accordingly.

3. Eva luation. The binary tree is evaluated by recursively traversing all its nodes. The Evaluate procedure is smart; it does not allow invalid operations, such as dividing by zero or finding the log of negative numbers.

(continued)

Points of Interest

Before CALC exits, it does three important things:

1. It calls the procedure FinishUp, which calls DestroyHeap. DestroyHeap frees up the memory the initialization code of the Storage module allocated to the 20K heap.

2. If the CALC window was opened, it closes it.

3. If CALC was launched from the Workbench, it calls the procedure ReturnToWB, which replies to the message the Workbench sent it upon start-up. If CALC failed to reply, the memory allocated to the message would be lost until a reboot.

When a procedure in the Interpreter module encounters an error, it calls a general purpose error routine which is provided through the procedure variable HandleError. The nice thing about this arrangement is that HandleError can be several different procedures, depending on which program imports from the Interpreter module, yet the Interpreter doesn't have to concern itself with these details. This is another example of Modula-2's flexibility.

The Evaluate procedure does not always avoid invalid or out-of-range conditions successfully. An out-of-range or invalid condition causes a run-time error, which then causes CALC (and often the Amiga) to crash. The TDI implementation solves this problem: You can trap and handle run-time errors yourself (This works properly only under version 3.00 of the compiler.). One of the first things CALC does is point the procedure variable ErrorProcessor (imported from AMIGAX) to a procedure which will shut things down and exit in an orderly fashion. When a run-time error occurs, procedure ErrorTrapper gets called, and it closes everything down and exits.

Compiling the Program

You should have no problems if you first compile all the definition modules (the ".Def" files) in this order:

1. MyInOut
2. MyRealOut
3. Interpreter0
4. Interpreter
5. WBStart

After all the definition modules are compiled, you can compile the six ".Mod" files.

All the modules should compile cleanly using version 3.00 of the TDI compiler. Compiled under version 3.00, CALC should work correctly under AmigaDOS 1.1 and 1.2. If you use version 2.00 of the compiler, you might have problems running CALC under AmigaDOS 1.2.

A Word About Style

My prime objective in porting this program to Modula-2 was to get it to work, not to make it look pretty. Also, I had to make the source fit into the 60-column width restriction imposed by typesetting requirements. Please keep that in mind if you come across a segment of code you consider hideous, or if you spot a blatant case of 'Pascal-ism' (a piece of code, obviously written for Pascal, which doesn't take advantage of the more advanced features of Modula-2). I apologize in advance.

About the Listings

I wanted to present the program already tailored to the new Benchmark Modula-2 compiler, but I ran into some unexpected problems (bugs) in the ported code. Therefore, the accompanying listings are for the TDI package only. In the future, however, I will present more and more programs for the Benchmark package, as I find it easier and faster to use. The accompanying listings include only some of CALC's modules; the rest will appear next month.

Suggested Reading

You can find out more about scanning and parsing in any good book on compiler construction, such as *Compilers: Principles, Techniques, and Tools* By Aho, Sethi & Ullman. Publisher: Addison Wesley.

More information on binary trees and recursion can be found in books dealing with data structures and algorithms, such as:

Algorithms + Data Structures = Programs By N. Wirth. Publisher: Prentice-Hall.
Data Structures Using Pascal By Tenenbaum & Augenstein. Publisher: Prentice-Hall.
Programming in Modula-2 By N. Wirth. Publisher: Springer-Verlag.

Some Late Breaking News

The Amiga Modula-2 compiler market keeps getting more and more interesting. Another contender joined the race: Interface Technology has a fast, one-pass compiler which has some interesting features (like handling run-time errors in a civilized manner). It lists for $199. I hope to have more to say about it next month. For more information, contact: Interface Technology, 3336 Richmond, Suite 323, Houston, Tx 77098 (713) 523-8422.


```
DEFINITION MODULE MyRealOut;
      (*********************************************)
      (*            (c) Copyright 1986, 1987       *)
      (*              by Steve Faiwiszewski        *)
      (*                                           *)
      (*    For non-commercial, non-profit use only.   *)
      (*********************************************)

PROCEDURE WriteReal(r : REAL; decimal: CARDINAL);
(* r       : The REAL to print                     *)
(* decimal : the number of digits printed after the *)
(*           decimal point.                        *)

END MyRealOut.
```

```
DEFINITION MODULE MyInOut;


PROCEDURE ReadString(VAR s: ARRAY OF CHAR);
(* Read a string of any characters, including blank *)
(* Reading is terminated when 's' is filled or EOL  *)
(* is encountered.                                  *)

END MyInOut.
```

```
DEFINITION MODULE Interpreter0;
 (*********************************************************)
 (*                                                       *)
 (*            (c) Copyright 1986, 1987 by                *)
 (*          Steve Faiwiszewski & Richie Bielak           *)
 (*                                                       *)
 (*        For non-commercial, non-profit use only.       *)
 (*                                                       *)
 (*********************************************************)

CONST
     maxtree = 40;
     maxoper = 40;

TYPE
     nodep = POINTER TO node;
     node = RECORD
               lson : nodep;
               val  : REAL;
               oper : CHAR;
               rson : nodep
            END;

VAR
     treestack : ARRAY[1..maxtree] OF nodep;
     operstack : ARRAY[1..maxoper] OF nodep;

PROCEDURE WriteLine(VAR line: ARRAY OF CHAR);
(* A WriteString followed by a WriteLn *)

PROCEDURE Trim(VAR sss: ARRAY OF CHAR);
(* Remove trailing blanks from a string *)

PROCEDURE FreeNodes(VAR t: nodep);
(* Dispose of the binary tree *)

END Interpreter0.
```

(continued)

```
DEFINITION MODULE Interpreter;
 (*******************************************************)
 (*                                                     *)
 (* Original concept implemented in Pascal on a PDP-11  *)
 (*                  by Richie Bielak                   *)
 (*          (CIS: 75716,352.  PLink: RICHIEB).         *)
 (*                                                     *)
 (* Enhanced and adapted to work with TDI Modula-2      *)
 (*          compiler by Steve Faiwiszewski             *)
 (*         (CIS: 74106,425. PLink: THE INTERN).        *)
 (*                                                     *)
 (*            (c) Copyright 1986, 1987 by              *)
 (*          Steve Faiwiszewski & Richie Bielak         *)
 (*                                                     *)
 (*       For non-commercial, non-profit use only.      *)
 (*                                                     *)
 (*******************************************************)

FROM CommandLine  IMPORT CLStrings;
FROM Interpreter0 IMPORT nodep;


TYPE
     CharSet = SET OF CHAR;
     tokentype = (func,operator,number,endofline,oops);
     tokenset  = SET OF tokentype;
     exptype = (assg,expr);
     ErrorType = (MissingValueError,
                  MissingOperatorError,
                  MissingOperandError,
                  MissingOpenParenError,
                  MissingCloseParenError,
                  IllegalValueForTanError,
                  IllegalValueForLogError,
                  IllegalValueForSqrtError,
                  IllegalValueForExpError,
                  DivideByZeroError,
                  GeneralError);

VAR
     tstack     : CARDINAL;
     ostack     : CARDINAL;
     whatwas,
     what       : tokentype;
     symbtable : ARRAY[0..26] OF REAL;
     valdef     : ARRAY[0..26] OF BOOLEAN;
     HandleError : PROCEDURE (ErrorType,CARDINAL);

PROCEDURE GrowTree(VAR line: CLStrings;
                   VAR LineIndex: CARDINAL);
(* Build the binary tree representing the expression *)

PROCEDURE Evaluate(Root: nodep; VAR suc: BOOLEAN): REAL;
(* Evaluate the binary tree *)

PROCEDURE InitStack;

PROCEDURE FinishUp;

END Interpreter.
```

•AC•

LIFE PART I: The Beginning

by Gerald Hull
People Link: DRJERRY

In 1970 mathematician John Horton Conway came up with a mathematical recreation based upon cellular automata theory he called "The Game of Life." It generates a special universe of objects which exhibit complex, unpredictable behaviors. In addition, the objects illustrate principles being applied today to fields as diverse as cosmology, particle physics, and thermodynamics.

As you will see, the Amiga is unique among all popular microcomputers in its ability to play Life. In the early days of the Game, it often took months, or even years, to uncover the properties of certain configurations, and to confirm or deny theories concerning their behaviors. Today, with the Amiga, it is possible to achieve the same results in under a minute.

Another von Neumann Architecture

John von Neumann is usually associated with the serial, stored program, single CPU architecture found in most of today's computers. In consequence, HyperCubes, neural nets, and other special parallel processing systems are often lumped together as "non-Von Neumann architectures."

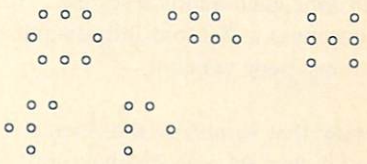
This categorization is ironic because von Neumann was also one of the first to investigate the properties of a particularly rich approach to parallel processing. This "cellular automata theory" studies the properties of systems composed of arrays of processors (or cells) representing independent finite state automata. Although each cell operates according to the same set of rules, they reach their own separate conclusions, based upon input. Usually that input consists of the states of the other cells in the neighborhood.

Von Neumann's interest was in the "self-healing" robustness of such systems of the multiple, independent units of such systems, and in the possibility of using them to demonstrate the mechanisms underlying self-reproduction. He was able to show that a cellular automaton consisting of 200,000 cells, each of which could be in one of twenty-nine different states, could reproduce itself. His proof anticipated the DNA mechanism of reproduction, discovered a few years later by Watson and Crick.

Conway's Game of Life uses much simpler automata. Its cells can be imagined as the squares of an infinite checkerboard. The neighborhood of each square consists of the eight squares immediately surrounding it. The rules governing these cells are deceptively simple. At any particular time, each cell is in one of two states: "alive" or "dead." In the next succeeding generation (or "tick"), a living cell remains alive only if it has two or three living neighbors, and a dead cell changes to living ("is born") only if it has exactly three living neighbors.

So this is not a "game" in the same sense as *Chess* or *Marble Madness*. There are no winners or losers, nor any special goal or payoff. The point, rather, is to see how the repeated application of this simple set of rules can produce an amazing, unpredictable universe of objects and behaviors.

For example, the following are five of the more common Life objects.



Of the five objects, one will disappear completely in six generations. One will go through a four tick cycle that results in the same pattern, but shifted one square diagonally (so that it "glides" across the screen).

After sixteen generations, another pattern will settle down into a symmetrical "still life," or unchanging pattern. One of the objects is an "oscillator" with a period of two: it repetitiously cycles between two separate configurations. Finally, one will explode into an incredible profusion of different objects and debris that won't settle down until 1,103 generations have passed. Which is which? The only way to find out is to apply Conway's rules over and over again—"recursive reiteration"—and to see what happens.

(continued)

Life on the Amiga

I don't intend to go any deeper here into the phenomenon of the Game of Life. There are a number of excellent articles and books on the subject, some of which are listed in the bibliography accompanying this article. What I'd like to turn to is the excellent advantage of exploring the Life universe with an Amiga.

The first substantive program I wrote in C was an implementation of Life that I sent off to John Foust for the AMICUS public domain disks. He seemed unimpressed by my Life program. I found out why when I learned of a package called the "Gizmoz Productivity Set." It had a Superlife program with a 320-by-180 cell mode that raced along at more than four generations a second. This represented a 300 fold improvement over my poor version!

I knew that somehow the Amiga's special chip set was involved, but didn't learn the details until I joined the PeopleLink (PLink) network. A programmer named Scott Evernden had uploaded source code that showed how to use the Amiga's blitter to implement the Game of Life. His algorithm was derived from a *Byte* magazine article by Mark Niemiec (see bibliography).

Another PLink user named Alonzo Gariepy soon uploaded a version of the Game which galloped a 320-by-200 cell array along at seventeen generations per second! It had taken Everndon 39 blits to calculate a single Life generation, and Gariepy did the job in just 10! With justifiable pride, he stated, "As of this writing, I believe this is the fastest micro implementation of Life." Alas, Gariepy did not post his source code.

I tried to figure out Gariepy's solution for a number of months, but I couldn't seem to reduce the Life algorithm to only ten blits; the best I could do was eleven. Finally, I chanced yet another version of the Game of Life by Tomas Rokicki on Fred Fish Disk #31. There it was at last—the ten blit solution. Astonishingly, I realized that the trick I had overlooked would simplify my approach by two more blits.

More by stubbornness than ingenuity, I had done Gariepy one better. I had a nine blit solution to the Game of Life! In Part II of this article, I will publish the complete source to this nine blit version. Until then, readers who enjoy mathematical puzzles can see if they can figure it out on their own. Shortly, I will describe the features of the blitter that are relevant to the solution.

Using the Blitter for Graphics

First, let's see why a computer designed for special graphics capabilities has a fantastic talent for cellular automata. The chart I have included shows that Amiga versions of the Game of Life are more than 300 times faster than highly optimized IBM PC programs. I am tempted to suggest that this simply reflects the relative worth of the two machines, but a more detailed answer is possible.

Most Amigans know about the blitter in connection with "bobs," or blitter objects. Although bobs are slower and tend to use more memory than sprites, they can be any size and can have as many colors as the screen. By giving the blitter maximum flexibility for manipulating bobs, its designers created a chip with much broader powers.

The basic function of the blitter is to move a block of data from one area of memory to another. Let's see how this can be used for graphics animation. Suppose you have a game in which you want to move a foreground object, say a car, across a background of road and houses (Any resemblance between this description and material in Chapter Six of the *AMIGA HARDWARE REFERENCE MANUAL* is hardly coincidental.).

Here's one way to do it. You start with a screen that has just the background on it (call it A) and another with just the car (B). These screens can consist of one to five bitplanes, but it's simplest to think of them as one bitplane deep. Now make make a third stencil screen (C) with 1's everywhere the car screen has a 0, and 0's everywhere it has a non-0 value.

We can create a new screen by logically ANDing the background screen with the stencil screen. This action creates a screen with a hole in it the shape and size of the car. Next, OR this screen with the car screen, and *voila*, you have the car set against its background. Or, symbolically:

```
D := A and C;
D := D or B;
```

To make the car move, all you need to do is shift both the car and car stencil screens in the appropriate direction by an appropriate amount, and repeat the process. The result is a car moving against a backdrop of road and houses.

The Amiga blitter has the ability to perform all these operations in a single step, or "blit." It will work with up to three source bitplanes (A, B, and C) for each destination bitplane (D), and two of the three (A and B) can be shifted by any arbitrary amount. In addition, the blitter can render the destination as any logical function of the three source bitplanes.

Operations that produce an object-on-background, like our car-and-road example, are used so often that they have acquired a special name. Here's such a "cookie cut" blit for the above:

```
D := (A and C) or B;
```

which is logically equivalent to:

```
D := (A and B and C)
or (A and B and not-C)
or (A and not-B and C)
or (not-A and B and C)
or (not-A and B and not-C);
```

The latter version of the formula—something that logicians call "disjunctive normal form"—tells us how to encode this particular logical function for the blitter. A special eight-bit value codes which of the possible states of the three source bitplanes will produce a pixel in the destination bitplane. To perform our particular cookie-cut, we use the code 0xEC = 11101100.

	A	B	C
1	1	1	1
1	1	1	0
1	1	0	1
0	1	0	0
1	0	1	1
1	0	1	0
0	0	0	1
0	0	0	0

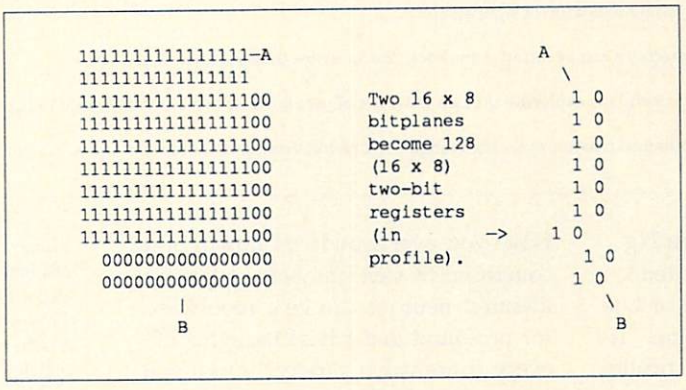
Cellular Automata with Blits

The same blitter functions that make the Amiga such a facile manipulator of graphics also give it a formidable capacity for cellular automation. Many readers will recall the task of building a one- or two-bit adder from simple logic circuits. Now think of a bitplane as a big sheet of registers; two bitplanes give you two-bit registers (turn them sideways in your imagination), and so forth.

Instead of logic chips, you simply use blitter operations in the appropriate sequence. For example, the result, in low resolution mode, is a big sheet containing 320 x 200 separate sums. This is the blitter's strength for simulating cellular automata. It can perform the same calculation simultaneously on as many separate cells as there are pixels in a bitplane, all at DMA rates.

Thus, though the blitter is not "really" a parallel processor (It doesn't consist of a number of separate CPUs.), it can be programmed so that it functions like such a processor. How does it compare with special hardware designed specifically for cellular automata? Here's an instance of the latter.

"The Cellular Logic Image Processor number 4 (CLIP4) ... is an array of 9216 one-bit CPUs .... When conducting more complex nearest- neighbor operations, its speed is equivalent to over ten billion instructions per second. This makes CLIP4 one of the fastest computers in the world .... It will be produced by commercial manufacturers in 1984 at a price of only 100 thousand dollars." (*MODERN CELLULAR AUTOMATA*; see bibliography.)



By comparison, it takes the Amiga about 5 milliseconds to blit a lo-res destination from three source bitplanes. Counting each bitplane separately, we get 3 * 200 * 320 * 200 = 38 million operations per second on a computer than can be bought for under $1000. I wonder if there are any good games for the CLIP4?

Let's see how the Amiga blitter enables us to calculate a generation of the Game of Life. A low resolution, monochromatic Amiga screen (just one bitplane) is generated from an 8000 (320 * 200 / 8) byte block in chip memory—each bit equals one pixel. For Life, a value of one in that bit represents a living cell, and zero means a dead cell.

In order to make the calculations, you allocate some additional bitplanes to serve as registers for our calculations. By using the right shifts and offsets, you can use the display screen as source A, the same screen shifted one pixel to the left as source B, and now shifted one pixel to the right as source C.

For destination, you can use one bitplane register to hold the first bit of the sum, and another to hold the carry. With repeated applications of this procedure, you simply add up the number of "on" pixels in the "neighborhood" of any given pixel. In virtue of the blitter, the same calculation is performed on all the pixels on the screen at the same time.

Next, apply the Life rules to those sums, and generate a new bitplane in which a pixel is on—that is, its bit is set—if, and only if, it: (a) was on during the previous generation and had two neighbors; or (b) had three neighbors. This solution bitplane becomes the new display screen. The Amiga's blitter has calculated the next Life generation.

The Meaning of Life

That should be enough information for anyone who wants to figure out how one generation can be calculated in just nine blits. The concluding article in the Life pair will discuss that algorithm in detail, along with procedures for manipulating the Amiga blitter to perform such tasks. Not everyone is fascinated by mathematical puzzles and games, though. So let me finish with a few remarks on the broader implications of Life.

One of the problems of cosmogony—the scientific study of the beginning of the universe—is explaining how the lavish diversity of existence could arise

(continued)

LIFE PROGRAMS ON THE AMIGA AND IBM PC

	MICRO	LANG	MODE	ARRAY	GEN/SEC	CELL*GEN/SEC	*IBM (a)
POUNDSTONE(b)	IBM PC	BASIC	B	20x38	1/15	51	0.01
HULL	Amiga	C/Asm	B	38x38	1/2	800	0.18
POUNDSTONE(b)	IBC PC	Asm	W	50x80	1.1	4400	1.00
GIZMOZ(c)	Amiga	C?	W	320x180	4.3	247680	56.29
EVERNDEN	Amiga	C	B	314x187	5.3	311205	70.73
GARIEPY	Amiga	C	W	320x200	17.2	1100800	250.18
ROKICKI	Amiga	C	B	318x198	19.8	1246687	283.34
HULL	Amiga	C/Asm	W	320x200	19.9	1273600	289.45
——(d)	Amiga	——	B	318x198	23.2	1460765	331.99

(a) The two different modes are bordered and wrap-around (or "toroidal"). With a wrap-around screen, a glider sailing off the top of the screen appears at the bottom. The same holds true for right and left.

(b) William Poundstone's programs can be found in his book *The Recursive Universe* (see bibliography.).

(c) Gizmoz's SuperLife, very useful for exploring the Life universe, allows the user to control the speed of execution.

(d) This is the current hypothetical maximum for the Amiga—a nine-blit version without the elegant, but blit-consuming wrap-around feature.

from a simple beginning, like the Big Bang. Was it all somehow encoded right from the start? The Game of Life suggests an alternative explanation. It shows how complex and varied results can emerge from simple beginnings governed by simple rules.

The science of thermodynamics has problems of its own. According to the famous Second Law, the amount of disorder—"entropy"—in the universe must steadily increase. How can one explain the emergence of structures of ever-increasing complexity? The Game of Life shows a way out of this dilemma by providing a model in which structures are inevitable by-products of random reactions.

If you are puzzled by claims that particles like protons and neutrons are made up of even more fundamental particles called quarks, you might find a helpful analogy in the way a peculiar arrangement of Life cells can come together as a "glider," which thereupon takes on a "life" of its own.

Have you ever wondered how a brain composed of vast numbers of largely identical neurons can be a repository for profound and trivial thoughts of every conceivable variety? Once you become familiar with the odd and intricate objects that populate the Life universe, you will acquire a great respect for the potential conglomerations of independent processing units.

What's Life? You've read the magazine, now play the Game! With your Amiga computer you have an "unparalleled" opportunity for investigating the curious and fascinating universe of the Game of Life. Happy exploring!

A Brief Bibliography for the Game of Life

P.W. Atkins, *The Second Law*, Chapter 9. Scientific American Books, 1984.
A very accessible discussion of the light shed upon the mysteries of thermodynamics by the Game of Life.

Arthur W. Burks, ed., *Essays On Cellular Automata*. University of Illinois Press, 1970.
Not for the faint of heart or weak of mind; don't look at me, I haven't read it!

Martin Gardner, *Wheels, Life And Other Mathematical Amusements*, Chapters 20-22. W.H. Freeman and Company, 1983.
This work, by *Scientific American's* legendary Mathematical Games columnist, includes the articles that first introduced the Game to the public at large.

Mark D. Niemac, "Life's Algorithms." *Byte*, January 1979.
A nice survey of different algorithms for the Game of Life and some of its more common variations.

Kendall Preston, Jr., and Michael J.B. Duff, *Modern Cellular Automata*. Plenum Press, 1984.
If you really want to fry your brain with the mathematics of cellular automata theory and some of its practical applications, you might check this one out.

William Poundstone, *The Recursive Universe*. William Morrow and Company, 1985.
This is perhaps the best general introduction to the Life universe in all its glorious detail. The Game's significance, not just for thermodynamics and cosmology, but also for particle physics and the ongoing search for a Grand Unified Theory (GUT) is outlined.

•AC•

AmigaNotes

Music on the New Amigas

by Rick Rae
CIS# 76703,4253

As I write this (mid-October), the Amiga 500 has been selling for some time and the 2000 is just becoming consistently available in computer stores. With this in mind, it might be worthwhile to take a look at making music with the new machines. As you might expect from the level of compatibility between the new entries and the 1000, there are only two significant differences at this time.

The New Anti-Aliasing Filter

Many people aren't aware that the output filter on the new machines can be bypassed. This filter, which is always active on the 1000, has a corner frequency of about 4 KHz and a cutoff slope of perhaps 40 db per octave. In audiophile terms (where frequency response is normally quoted at the 3 db points), the Amiga has a top end of considerably less than 5 KHz. Even if you stretch to the limit, 7 KHz would have to be the maximum. Anyone who knows audio will readily admit that this limitation doesn't do much for sound quality: It's akin to turning your amplifier's treble control all the way down.

A few hardware hacks opened their machines and bypassed this filter, but most of us just lived with it. With the 500 and 2000, however, this change is no longer necessary. Commodore has added control circuitry that allows the filter to be bypassed, by software!

Whenever new circuitry is added to an upward-compatible system, there's always the question of where to put it. Commodore found a tidy solution: The control line for the Power LED also switches the filter. This duality means the system comes up with a bright Power indicator and the filter enabled. Turn on bit PA1 of CIA A (the same 8250 whose other port drives the parallel interface), and the Power indicator goes to half intensity as the filter opens up. Not only does this scheme avoid any incompatibility problems, it even indicates the state of the filter.

(A note of caution: I am told that the new circuitry is included in the "Fat Agnes" custom chip. If this is the case, the German version of the 2000 doesn't have the switchable filter.)

It would be fairly simple to cobble up a small program to toggle this bit, but if you're not a programmer, don't despair; it's already been done for you. Gregg Tavares [70275,627] has written a program called FILT, which is available from CompuServe's AmigaForum under the name DL15:FILT.ARC. This is a handy little mouse-driven utility with its own window—handy, since you can lay FILT's small window over your Pro MIDI or DMCS or whatever screen, and have instant access to the controls without flipping screens.

Don't expect elimination of the filter to make your Amiga sound like a $10,000 dedicated sampler. For one thing, we're still limited by the 8 bit S/N ratio; for another, bypassing the filter means aliasing can occur on the output, resulting in all sorts of buzzes and whines and tweety birds if you're playing back a sample with lots of high frequency information. Still, it's a step in the right direction and opens up some new possibilities. If you have a 500 or 2000, grab a copy of FILT and play with it.

Problems with MIDI Interfaces

Barry Massoni [73260,1413] brought this problem to my attention. His MIDI interface would not run on his Amiga 500. I knew the gender of the serial port had been changed to match the IBM PC, but Barry already had a gender bender, so that wasn't his problem. Through Barry's research and the help of others on the forum, it soon became obvious that owners of the new machines have a problem.

The original 1000 serial port provided +/- 12 and +/- 5 volts, with the latter pair normally used to power MIDI interfaces. With the new models, Commodore not only switched the gender of the DB25 connector, but they also disconnected the 5-volt supply pins.

So the problem involves figuring how to provide the +/- 5 volts to the MIDI interface. One approach uses an external power supply, but most small calculator-type adapters only provide one voltage at a time; dual supplies are a bit scarce and more expensive. Barry did some calling around and thinking, and came up with a reasonable alternative: Regulate the 12 volt

(continued)

lines down to 5 volts and drive the MIDI interface with that. (See Barry's instructions on page 109.) The original instructions are available from the AmigaForum as DL15:MIDIAD.TXT. Although it was designed specifically for the Mimetics unit, it should work with any MIDI interface which doesn't draw too much current. I haven't tried it, but Barry says his solution works quite well.

Another approach saves you some money if you have a 2000. Jeff Arnold of Golden Hawk Technology tells me that the 26-pin header, located immediately behind the serial plug, contains all the signals, including the five volt lines. If you don't mind running a ribbon cable into the 2000, you could conceivably make up a custom cable to connect your older MIDI interface.

If you're not a hardware hacker, there are other alternatives. Once I was aware of the problem, I called various interface manufacturers to find out what options are available. Here's what I found:

ECE R&D CORPORATION

Chuck Sanders informed me that ECE is finalizing a new MIDI interface called the "MIDI 500," designed specifically for the 500 and 2000. It will look exactly like the current model. The price will be the same as the original. If you have the old interface, you can exchange it for a brand new unit for $25.00.

GOLDEN HAWK TECHNOLOGY

Jeff Arnold notes that Golden Hawk is working on a completely new MIDI interface for the 500, but that it's a bit too early to quote a price or talk about upgrade policy. I'm not supposed to divulge the details I do have, so I'll just say it sounds like this new product will be the typically excellent Golden Hawk effort. They also have some interesting plans for the 2000 which, if they pan out, could save you quite a bit of money over buying a new interface. I'll pass along details as they firm up.

MIMETICS

Dave Rasmussen says the new Mimetics interface is already in production. Like the 1000 version (model HMC-1), it's built into the adapter cable. The only physical difference is the gender of the plug and the lack of a THRU port. The new model, HMC-2, can be used on a 1000 if you use the appropriate gender bender. Price remains the same and upgrade plans were not finalized at press time.

J. MICHAELS COMPANY

I was unable to contact J. Michaels Company to inquire about their interfaces. The original phone number has been disconnected, and I haven't been able to track down a new one. If anyone knows where I can contact them or what the situation is, I'd appreciate hearing from you.

SKYLES ELECTRIC WORKS

These folks managed to miss out on my first MIDI interface roundup, but they do have a 1000 unit available, so I'll take a moment to give you a brief overview.

The unit from Skyles Electric Works provides one IN port, one THRU port, and two OUT ports. Rather than being mounted in an enclosure, the entire interface is built on a two-by-three inch printed circuit board, with strategically placed blocks of epoxy encasing the components. The DB25 connector is mounted on one side of the board, and the MIDI jacks are on the other side, so it's almost like having a built-in MIDI interface.

Unfortunately, this is not the best approach. Not only is it impossible to perform any sort of repair if the interface fails, but you have to reach around the back of the computer to swap MIDI cables. Anyone who has dealt with more than a minimal system knows that quite a bit of cable swapping goes on. This interface makes it more difficult than need be.

The interface instructions are so detailed that you could almost see them as a joke. Two folded sheets explain how you plug the interface module into the serial port. These instructions are prefaced with comments such as, "If you are concerned about 'getting your fingers into' your AMIGA computer, please have your local dealer install the MIDI For AMIGA." Plugging a device into the serial port is getting your fingers into your computer?!

Actually I shouldn't poke fun at this effort, even if it might be overkill. The instructions include annotated photographs and in-depth instructions, and *anyone* should be able to install the interface. If you are all thumbs, this could very well be the perfect interface for you.

I spoke with Bob Skyles about their new interfaces, which are already shipping. "MIDI for Amiga 500" and "MIDI for Amiga 2000" are similar to the original 1000 model, except that the module plugs into the Amiga horizontally instead of vertically (better for visibility). Rubber feet support the board when the MIDI cables are plugged in. The two new interfaces are identical except for the height of the rubber feet, which varies to match the machine. Retail price remains the same. No upgrade details were available, but Bob invites interested owners to call for further information.

THE AMAZING MIDI INTERFACE

What about our own interface ("AmigaNotes," V2.2)? Since the 6N138 is limited to an absolute maximum of seven volts, there really isn't any way to use the 12 volt lines. Probably the best approach would be to simply build Barry's circuit right into the box. If you do this, you can substitute straight wire for his 1N914 diodes. Theoretically, the 6N138 could be

MIDI Interface Comparison Chart

NOTES	MANUFACTURER	RETAIL	SIZE (HxWxD)	IN	OUT	THRU
	Do-It-Yourself		As you please!	2	2	1
Serial Passthrough	ECE R&D Corp.	$59.95	1.75x3.25x5.25	1	1	1
Serial Passthrough	Golden Hawk	$79.95*	1.40x5.70x3.25**	1	2	0
Sync Output	J. Michaels Co.	$49.95	1.50x3.75x4.20	1	1	1
	Mimetics	$49.95	Cable assembly	1	1	1***
	Skyles Electric	$49.95	3.00x2.10x1.20**	1	2	1

*These specifications for the 1000 models apply to the 500/2000 versions as well, except as noted: *Retail price has not been established. **Form factor has changed. ***No THRU jack available.*

replaced with a 6N139, which is good up to 18 volts. In practice, however, a complete redesign of the circuit would be required and would probably be more trouble than it's worth.

The New Interface Review

Since the Skyles Electric interface didn't make it into the first roundup, I've included a new table for those of you shopping for a MIDI interface.

You might want to refer to the earlier article ("AmigaNotes," V2.1) for detailed information, but please ignore the original chart; some errors crept in during printing.

That's going to do it for now. I'm going to turn the rest of the column over to Barry Massoni (See page 109.). As for me ... See you next month!

•AC•

ECE Research and Development Corporation
1651 North Monroe Street
Tallahassee, FL 32303
904-681-0786

Golden Hawk Technology
427-3 Amherst Street Suite 389 Nashua, NH 03061
603-882-7198

J. Michaels Company
2232 Summit Street
Columbus, OH 43201
(*Status unknown*)

Mimetics, Inc.
PO Box 60238, Station A
Palo Alto, CA 94306
408-741-0117

Skyles Electric Works
231-E South Whisman Road
Mountain View, CA 94041
800-227-9998
415-965-1735

Part III

Animation for C Rookies

Double Buffered Animation Objects

by Michael Swinger

In the first two articles of this series, I have shown some relatively simple ways of using bobs to do traditional cel animation on the Amiga. The first program illustrated the routines necessary to display bobs. The second program used the system animation routines to turn the bobs into Animation Objects. This last article addresses the problem of double-buffering, which is necessary for any elaborate animation.

Double-buffering involves creating 2 BitMaps for your screen and displaying them alternately. This operation is a little more cumbersome, and your programs will run more slowly because of the time used by the RethinkDisplay() routine. Double-buffering also uses more precious Chip memory because you not only have to use 2 BitMaps for a screen, but you also must create a second SaveBuffer to save the background for the bobs in the second BitMap.

Double-buffering has not been very well documented by Amiga literature. The *RKM* mentions what you need to do, but the sample animation program at the end of the chapter on graphics does not use this information. A double-buffered sample program based on the *RKM* example is in the public domain, but it suffers from the "developer turgidity" that is so frustrating to beginning programmers. Other Commodore sample programs have used the idea of swapping multiple BitMaps in and out (such as in the Dual Playfield examples). I have borrowed a few ideas from these programs.

The clearest and most concisely written C code I have seen appears in a public domain program called "3D-Arm" by Bob Laughlin. It is on a Fish disk and I encourage you to marvel not only at the program, but at the clean and logical source code. Mr. Laughlin uses double-buffering, and, as he says, "It is simple, but not obvious."

For the benefit of those who are typing these programs, I have included the bob data in all examples. As your programs become more elaborate and the data becomes longer, you will probably want to put your data in a separate file, compile it separately from the main program, and then link the files later.

If you are using the gi program to create bob data, you can strip out much of the commented information and use the AmigaDOS JOIN command to join all your files. JOIN involves a lot of noisy disk grinding (especially if you have an early Amiga with the Vegematic drives), so you will probably want to join all the files in RAM: and then write them to a disk.

A typical data file would look like the following:

```
#include <Exec/types.h>
WORD Image_data1[130) = {
  ** data here for bob #1 **};
WORD Image_data2[130) = {
  ** etc. ** };
```

In place of the bob data at the beginning of your program, you would have statements like the following:

```
EXTERN WORD Image_data1();
EXTERN WORD Image_data2();    etc.
```

Be sure to write down the sizes of your bobs as you edit the data files. If you do this, you don't have to search a separate source file for the information of the buffers and VSprite structures. You could also include the data for the colormap in this file, as this data is unlikely to change.

Program #3
Double Buffered Animation Objects

```
/*** NOTE  There is a close gadget
on the window to end the program,
but it must be wiped out with the
SetRast call. Just keep clicking
in the upper left corner where you
think the gadget should be.
Some of the variable names have
changed from earlier programs.  We
are now using the Screen's RastPort
and ViewPort, rather than the
Window's structures.  ***/

#include <functions.h>   /* For
                         Manx only */
#include <intuition/intuition.h>
#include <graphics/gels.h>
struct  IntuitionBase *IntuitionBase;
struct  GfxBase *GfxBase;
struct  Screen *Screen1;
struct  Window *Window1;
struct  ViewPort  *VP1;
struct  RastPort  *RP1;
struct  AnimOb  Obj1,  *animKey;
struct  AnimComp  comp1,  comp2;
struct  GelsInfo  gelsinfo;
struct  VSprite  s1,  s2;
struct  collTable  Ctable;
VOID  Drawit();
/***   NOTE  1  ***/
struct  BitMap  BM[2];
SHORT  TOGGLE  =  0;
PLANEPTR  rastptr=NULL;
```

```
WORD  Image_data1[130]  =  {
   /*  Width:      18  (pixels)
       Height:    13  (pixels)
       Depth:      5   (planes)*/
   0x0000,0x0000,0x0000,0x0000,0x0080,
   0x0000,0x0080,0x0000,0x0080,0x0000,
   0x0080,0x0000,0x0080,0x0000,0x0080,
   0x0000,0x0080,0x0000,0x0080,0x0000,
   0x0000,0x0000,0x0000,0x0000,0x0000,
   0x0000,0xffff,0x8000,0xffff,0x8000,
   0xfc1f,0x8000,0xfc1f,0x8000,0xfc1f,
   0x8000,0xfc1f,0x8000,0xfc1f,0x8000,
   0xfc1f,0x8000,0xfc1f,0x8000,0xfc1f,
   0x8000,0xffff,0x8000,0xffff,0x8000,
   0xffff,0x8000,0x0000,0x0000,0x0000,
   0x0000,0x03e0,0x0000,0x03e0,0x0000,
   0x03e0,0x0000,0x03e0,0x0000,0x03e0,
   0x0000,0x03e0,0x0000,0x03e0,0x0000,
   0x03e0,0x0000,0x0000,0x0000,0x0000,
   0x0000,0x0000,0x0000,0x0000,0x0000,
   0x0000,0x0000,0x0360,0x0000,0x0360,
   0x0000,0x0360,0x0000,0x0360,0x0000,
   0x0360,0x0000,0x0360,0x0000,0x0360,
   0x0000,0x0360,0x0000,0x0000,0x0000,
   0x0000,0x0000,0x0000,0x0000,0x0000,
   0x0000,0x0000,0x0000,0x0000,0x0000,
   0x0000,0x0000,0x0000,0x0000,0x0000,
   0x0000,0x0000,0x0000,0x0000,0x0000,
   0x0000,0x0000,0x0000,0x0000,0x0000,
   0x0000,0x0000,0x0000,0x0000,0x0000 };

WORD  Image_data2[130]  =  {
/*     Width:      18
       Height:    13
       Depth:      5   */
   0x0000,0x0000,0x0000,0x0000,0x0000,
   0x0000,0x0000,0x0000,0x0000,0x0000,
   0x0000,0x0000,0x0000,0x0000,0x0000,
   0x0000,0x0000,0x0000,0x0000,0x0000,
   0x0000,0x0000,0xffff,0x8000,0xffff,
   0x8000,0xffff,0x8000,0xfc7f,0x8000,
   0xfbbf,0x8000,0xf7df,0x8000,0xffbf,
   0x8000,0xff7f,0x8000,0xfeff,0x8000,
   0xfdff,0x8000,0xfbff,0x8000,0xf00f,
   0x8000,0xffff,0x8000,0xffff,0x8000,
   0xffff,0x8000,0x0000,0x0000,0x0000,
   0x0000,0x0000,0x0000,0x0000,0x0000,
   0x0000,0x0000,0x0000,0x0000,0x0000,
   0x0000,0x0000,0x0000,0x0000,0x0000,
   0x0000,0x0000,0x0000,0x0000,0xffff,
   0x8000,0xffff,0x8000,0x0000,0x0000,
   0x0380,0x0000,0x0440,0x0000,0x0820,
   0x0000,0x0040,0x0000,0x0080,0x0000,
   0x0100,0x0000,0x0200,0x0000,0x0400,
   0x0000,0x0ff0,0x0000,0x0000,0x0000,
   0xffff,0x8000,0xffff,0x8000,0x0000,
   0x0000,0x0000,0x0000,0x0000,0x0000,
   0x0000,0x0000,0x0000,0x0000,0x0000,
   0x0000,0x0000,0x0000,0x0000,0x0000,
   0x0000,0x0000,0x0000,0x0000,0x0000,
   0x0000,0xffff,0x8000,0xffff,0x8000 };

WORD  Sbuffer1[2*13*5];
WORD  Cmask1[2*13];
WORD  Bline1[2];
/***  NOTE  2  ***/
WORD  Dbuffer1[2*13*5];
WORD  Sbuffer2[2*13*5];
WORD  Cmask2[2*13];
WORD  Bline2[2];
WORD  Dbuffer2[2*13*5];

USHORT  colormap  [32]  ={
   0x0ccd,0x0685,0x0f79,0x0f30,0x0f9b,
   0x0f90,0x0eb5,0x0fc8,0x0eef,0x0ddd,
   0x0ccc,0x0aaa,0x0fdc,0x0fcb,0x0fba,
   0x0ea9,0x0e93,0x0d90,0x0c87,0x0a54,
   0x0c75,0x0777,0x0556,0x0069,0x09e1,
   0x0fed,0x07c1,0x05a0,0x0270,0x0fdd,
   0x0fff,0x0000 };
```



```
struct  VSprite  v1  ={
   NULL,NULL,NULL,NULL,NULL,NULL,
    OVERLAY  |  SAVEBACK,
   10,20,13,2,5,0,0,&Image_data1[0],
   &Bline1[0],&Cmask1[0],NULL,NULL,
    0x01f,0,NULL  };

struct  DBufPacket  Db1  ={  NULL,NULL,
   NULL,&Dbuffer1[0]};

struct  Bob  b1={
 BOBISCOMP,&Sbuffer1[0],&Cmask1[0],NULL,NULL,
  &v1,&comp1,&Db1,NULL};


struct  VSprite  v2  ={
   NULL,NULL,NULL,NULL,NULL,NULL,
    OVERLAY  |  SAVEBACK,
   90,50,13,2,5,0,0,&Image_data2[0],
   &Bline2[0],&Cmask2[0],NULL,NULL,
    0x01f,0,NULL  };

struct  DBufPacket  Db2  ={  NULL,NULL,
   NULL,&Dbuffer2[0]};

struct  Bob  b2={
 BOBISCOMP,&Sbuffer2[0],&Cmask2[0],NULL,NULL,
  &v2,&comp2,&Db2,NULL};


struct  NewScreen  NewScreen1  ={
   0,0,320,200,5,1,0,
    NULL,CUSTOMSCREEN  |  CUSTOMBITMAP,
    NULL,NULL,NULL,&BM[0]  };
```

```
struct  NewWindow  NewWindow1  ={
   0,0,320,200,1,0,CLOSEWINDOW,
    SMART_REFRESH  |  ACTIVATE  |
    BORDERLESS  |  WINDOWCLOSE,
   NULL,NULL,NULL,NULL,NULL,0,0,0,
    0,CUSTOMSCREEN    };


main()
{
SHORT  x;

Open_Libraries();

Open_Screens();

Init_Bobs();

   for  (x=0;  x<500;  ++x)
   {  Drawit();
   }
Cleanup();

}     /***  end  main  ***/

Open_Libraries()
   {
    IntuitionBase  =  (struct
     IntuitionBase  *)
    OpenLibrary("intuition.library",0);

    GfxBase  =  (struct  GfxBase  *)
    OpenLibrary("graphics.library",0);
   return();
   }
```

(continued)

```
Open_Screens()
  {
   SHORT j;
/***NOTE 3 ***/
   InitBitMap(&BM[0],5,320,200);
   InitBitMap(&BM[1],5,320,200);
        for (j=0; j<5;j++)
        { BM[0].Planes[j] = (PLANEPTR)
           AllocRaster(320,200);
          BM[1].Planes[j] = (PLANEPTR)
           AllocRaster(320,200);
        }
   Screen1=OpenScreen(&NewScreen1);
   NewWindow1.Screen=Screen1;
   Window1=OpenWindow(&NewWindow1);


    RP1 = &Screen1->RastPort;
    VP1= &Screen1->ViewPort;
    #define RI   VP1->RasInfo
   Screen1->RastPort.Flags=DBUFFER;
   LoadRGB4(VP1,&colormap,32);
/*** NOTE 4 ***/
   RP1->BitMap=&BM[0];
   SetRast(&Screen1->RastPort,0);
   RP1->BitMap=&BM[1];
   SetRast(&Screen1->RastPort,0);
   RP1->BitMap=&BM[0];
   return();
}
  Init_Bobs()
{
   Obj1.NextOb=NULL;
   Obj1.PrevOb=NULL;
   Obj1.AnY=64*20;
   Obj1.AnX=64*10;
   Obj1.YVel=0;
   Obj1.XVel=0;
   Obj1.YAccel=0;
   Obj1.XAccel=0;
   Obj1.RingYTrans=3*64;
   Obj1.RingXTrans=2*64;
   Obj1.Headcomp=&comp1;
   comp1.Flags=RINGTRIGGER;
   comp1.Timer=50;
   comp1.TimeSet=10;
   comp1.Nextcomp=NULL;
   comp1.Prevcomp=NULL;
   comp1.NextSeq=&comp2;
   comp1.PrevSeq=&comp2;
   comp1.AnimCRoutine=NULL;
   comp1.YTrans=0;
   comp1.XTrans=0;
   comp1.HeadOb=&Obj1;
   comp1.AnimBob=&b1;

   comp2.Flags=RINGTRIGGER;
   comp2.Timer=50;
   comp2.TimeSet=10;
   comp2.Nextcomp=NULL;
   comp2.Prevcomp=NULL;
   comp2.NextSeq=&comp1;
   comp2.PrevSeq=&comp1;
   comp2.AnimCRoutine=NULL;
   comp2.YTrans=0;
   comp2.XTrans=0;
   comp2.HeadOb=&Obj1;
   comp2.AnimBob=&b2;

    gelsinfo.nextLine = NULL;
    gelsinfo.lastColor = NULL;
    gelsinfo.collHandler =NULL;
    Screen1->RastPort.GelsInfo =
&gelsinfo;
   v1.VSBob=&b1;
   v2.VSBob=&b2;
    InitGels(&s1, &s2, &gelsinfo);
   GetGBuffers(&Obj1,RP1,TRUE);
   InitGMasks(&Obj1);
   InitAnimate(&animKey);
   AddAnimOb(&Obj1,&animKey,RP1);
   return();
}
Cleanup()
{
    SHORT j;
   Wait(1<<Window1->
        UserPort->mp_SigBit);
    if (rastptr) FreeRaster
         (rastptr, 320,200);
   FreeGBuffers(&Obj1,RP1,TRUE);
   CloseWindow(Window1);
   CloseScreen(Screen1);
    for (j=0; j<5; ++j)
   {
     if (BM[0].Planes[j]) FreeRaster
          (BM[0].Planes[j],320,200);

     if (BM[1].Planes[j]) FreeRaster
          (BM[1].Planes[j],320,200);
   }
   CloseLibrary(GfxBase);
   CloseLibrary(IntuitionBase);
   return();
}
VOID Drawit()
/*** NOTE 5 ***/
    { RP1->BitMap =&BM[TOGGLE];
      RI->BitMap = &BM[TOGGLE];
           Animate(&animKey,RP1);
           SortGList(RP1);
           WaitTOF();
           DrawGList(RP1,VP1)
           MakeScreen(Screen1);
           RethinkDisplay();
           TOGGLE ^=1;
    }
```

NOTE 1

These are new declarations. We are setting up an array of 2 bitmaps, and, as we will see later, the TOGGLE alternates between BitMap[0] and BitMap[1].

NOTE 2

This buffer saves the background for the second BitMap. If you are not using the system animation routines, and you are using simple bobs, remember that you will have to call RemBob() or RemIBob() for each of the BitMaps in order to remove the bob from the gel list.

NOTE 3

The order of these statements is important. Several spectacular crashes occurred when I was first running the program, even though it would compile. You must initialize and allocate the bitmaps before opening any screens or windows, since you have already told the screen to expect a CUSTOMBITMAP.

NOTE 4

The SetRast calls clear the screens to whatever color you specify—in this case, it is color 0. This is what is wiping out the window's close gadget. If you really want to preserve the borders and gadgets, you can use GIMMEZEROZERO.

NOTE 5

Since the TOGGLE has previously been set to 0, we are writing into the first BitMap. When we reach the end of this function, the TOGGLE is bitwise EORed, which sets it to 1. It will then continue to alternate between 0 and 1 each time this function is called.

In the *Intuition Reference Manual*, Robert Mical warns us against using RethinkDisplay() in too cavalier a fashion ... but nothing else works. RemakeDisplay() is an even more potent and ominous statement that calls MakeScreen() for every screen. I prefer to limit its scope and call it for just my one screen.

•AC•

WARNING: A New Computer Virus May Be Hazardous To Your Amiga.

So far, Amiga public domain software has been free of a phenomena common to other computers—a class of programs known as "Trojan horses" or "viruses." When run on your computer, these programs cause some form of insidious damage. The Amiga now has a virus. This type of program is called a "virus" because of the way it replicates and transfers itself to other disks, like the way a cold is spread by shaking hands.

A European group called SCA claims responsibility. It is yet unknown what the virus does, but an examination of the virus code shows the message "Something wonderful has happened. Your AMIGA is alive! And, even better, some of your disks are infected by a VIRUS!" There have been one or two reports of this message being printed at random times, but no confirmed reports of any damage done to disks.

[Ed note: The Amiga 2000 is most susceptible to attack, according to early reports. A major software company attributes major losses to the virus' work on the 2000. Amiga 500s and 1000s seem unaffected thus far.]

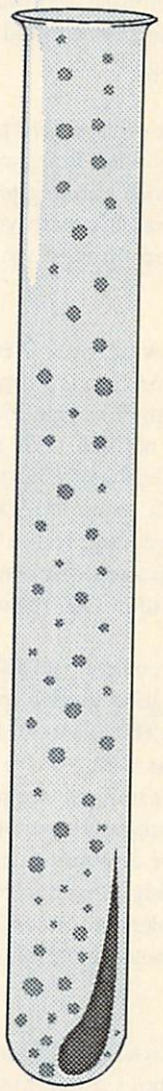
The virus is spread from computer to computer by an infected Workbench boot disk. If you start your Amiga with an infected disk, then the virus is transfered into memory. The virus survives a warm boot—that is, a restart of the Amiga by the "Vulcan nerve pinch," also known as CTRL-Amiga-Amiga. Any Workbench disk used to boot an infected Amiga will "catch" the virus if the write-protect tab is closed.

If you warm-boot with any Workbench after your computer has been "infected," then the virus will be copied to that new disk. Of course, booting with that disk at a later time will infect the machine again, and so on.

The virus code hides in the first block on a disk. Normally, this area holds the "boot block," or the very first piece of code loaded into the Amiga. The disk continues to work as usual. You can view the virus code with a disk editor, such as the public domain program "Sectorama."

There is a way to eradicate the virus from infected disks. The standard AmigaDOS INSTALL command rewrites a uninfected boot block to disk. By running INSTALL on all infected disks, then turning your machine off, you can eradicate the virus.

•AC•



Advanced 68000 Assembler Language Programming

The Big Picture

by Warren Ring

"How do you do a good job of writing a program? What do you keep in mind?"

I have always believed that assembly, on any machine, is the best language. By definition, it is the most efficient and fastest language. A good macro assembler, tuned with macros the programmer is comfortable with, can be used with as much ease and reliability as most high-level compilers, and with initial coding nearly as fast as a high-level language.

The element of assembly programming I like best is that you *know* exactly what you have. You can call your object code into RAM with a debugger, look at the disassembled object code, and match it with your source code. If necessary, you can even step through your program one instruction at a time.

Now, I know all your C friends are snickering, but when integration and debug time rolls around, and they're decorating their source code with "printf"s and wondering whether or not 00C07327 really points to a valid record, we'll see who has the last laugh. There are times when you have a really exotic bug, and you need to examine disassembled code to see where the misunderstanding, wrong language documentation, unwritten rule, or incorrect code generation lies, regardless of the source language.

In this first article on 68000 programming, we're going to talk about good general programming practices that apply not only to 68000 assembler programs, but to all other programs as well. We're going to discuss: (1) how to effectively comment programs; and (2) how to make assembly programs portable between machines incorporating the same or different microprocessors. You didn't know that assembly language programs can be portable between different microprocessors? Boy, are you green! They can be ported if they are set up correctly. I'll show you how this is done.

Before discussing 68000 assembly programming specifically, let's talk about programming principles in general, in any language. How do you do a good job of writing a program? What do you keep in mind? How should you approach it? What is the big picture? I once attended a two-day seminar put on by the Yourdon people. They had a great answer to this question. Studies have shown that over the life of a program, two thirds of the total effort that goes into writing done in the *maintenance* phase. Therefore, when you write a program, you should keep ease of maintenance in mind.

Ease of maintenance. What does that mean? It means you put in a little more effort at the beginning to save your hide later on. It means that you assume that two years from now, when you're called back to make a change, you can pull it out of your drawer and dust off the source listing without fear that essential pieces of information about the program have been long forgotten. You also do things *uniformly*. Don't use some procedures in one place and other procedures in other places to perform the same function. No quick-and-dirty procedures are allowed. Exercise the self-discipline it takes to do it right the first time, even if it means delaying introduction of the product.

Ease of maintenance also means anticipating the kinds of changes that may be needed later on. Put some hooks in for them, like array sizes and validity limits. My rule on this issue states that every value that is not a zero or one must be called out *symbolically* in the program and defined in a pool of equate (define) statements. No exceptions. (I don't always measure up on this one, but it's an ideal I always strive for.)

Comments

Now, let's talk about comments. I've been programming for 15 years, and I've discovered one thing—most programmers don't know much about commenting. Each time I get involved in a new programming situation, I notice the same mistakes. These mistakes occur because of a lack of planning. Lack of planning occurs because of inexperience. How do you get experience? I know, by making mistakes! After a losing season, famed Green Bay Packers coach Vince Lombardi once said , "Men, THIS is a FOOTBALL." Like

Lombardi, we're also going to cover some basics. We're going to talk about commenting in a style you may never have seen.

Let's talk specifically about comments. Ease of maintenance requires functional, but not physical, coupling of comments and source statements. In other words, a comment cannot reference a register, a cryptic variable name, or anything else that is machine or implementation dependent. How can that be? Comments should define *functions*, not machine attributes. Look at the following example; many of you will recognize this function. The routine converts a hex nybble to an ASCII character for display. The comments without the source code are:

```
DSP1HEX   ;"Display 1 hex character" routine
          ;(Push registers)
          ;Strip off all but bits 3-0
          ;If the value is less than 10,
          ; then jump to DSP1HEX1
          ;Add 7 to the value DSP1HEX1
          ;Add '0' to the value
          ;Display the character
          ;(Pop registers)
          ;Return
```

You must add the value of an ASCII zero to the value to convert a 4-bit nybble to a printable character. If the value started out between $0A and $0F, add 7 to put the value in the range of ASCII "A" through "F," since, on an ASCII conversion chart, there are 7 characters that reside between "9" and "A."

The "DSP1HEX1" is a label. Notice that there are no references to registers or variable names; there are references only to functions. Of course, there are no variables referred to here. If there were, I would require that they be referred to descriptively, rather than by their cryptic variable names. Let's assume we are programming on a 68000. The source code and comments should look like this:

```
DSP1HEX             ;"Display 1 hex character" routine
                    ;In: <D0> bits 3-0 = the value to display
MOVE.L D0,-(A7)     ;(Push registers)
ANDI.L #$0F,D0      ;Strip off all but bits 3-0
CMPI.L #10,D0       ;If the value is less than 10,
BMI    DSP1HEX1     ;   then jump to DSP1HEX1
ADDI.L #7,D0        ;Add 7 to the value

DSP1HEX1
ADDI.L #'0',D0      ;Add '0' to the value
JSR    DisplayChar  ;Display the character
MOVE.L (A7)+,D0     ;(Pop registers)
RTS                 ;Return
```


If you are programming an 8080 under CP/M, your source code and comments should look something like this:

```
DSP1HEX             ;"Display 1 hex character" routine
                    ;In: <A> bits 3-0 = the value to display
PUSH   PSW          ;(Push registers)
ANI    0FH          ;Strip off all but bits 3-0
CPI    10           ;If the value is less than 10,
JM     DSP1HEX1     ;   then jump to DSP1HEX1
ADI    7            ;Add 7 to the value

DSP1HEX1
ADI    '0'          ;Add '0' to the value
CALL   DSP1CHR      ;Display the character
POP    PSW          ;(Pop registers)
RET                 ;Return
```

If you happen to be programming an M-5, then the ingram programming would look like this:

```
DSP1HEX             ;"Display 1 hex character" routine
                    ;Args are passed in the usual manner
                    ;   described in 1672-820-443-2323.
MZ3>22A             ;(Push registers)
?3/Z,D0             ;Strip off all but bits 3-0
@000009QJ           ;If the value is less than 10,
JJS(ZQ/DSP1HEX1)    ;   then jump to DSP1HEX1
D0++++++++          ;Add 7 to the value

DSP1HEX1
ADD'0'//D0          ;Add '0' to the value
GORETDisplayChar    ;Display the character
MZ3<22A             ;(Pop registers)
RQGETRAPOPGOTO      ;Return
```

These comments are descriptions of functions, as they should be. Two years from now, when I'm looking through the source listings for a certain location in this program, I really won't care what registers are used. Rather, I'll care about which function is being carried out and where.

Did you notice that each comment is actually a complete sentence? Such construction should *always* be the case. The following examples should *never* be used as comments:

(continued)

No.
Is the loop done yet?
Get the next data value
Yes.

I consider the following examples acceptable:

Read the next disk record into the buffer
If the record type is invalid, then jump to GORF5
Wait here until a key is pressed
Display "Enter a value: "
If there was a read error, then jump to ERR003

Portability Between Machines With Different Microprocessors

Did you also notice that the comments in the above examples are machine independent? This structure leaves the door to portability wide open! You can move your source code to the new machine and go down the listing, adjusting the assembler source statements to match the comments' description. I have done this with great success, migrating line drawing routines from 8080-based CP/M machines to 6502-based Apple 2s.

The method used for commenting your assembler programs can have a dramatic impact on the maintainability of your programs. Now let's discuss portability of code between 68000 systems.

68000 Portability

We need to discuss one more aspect of portability specific to 68000 systems—the handling of system calls. Let's say you've written the world's greatest Amiga program. Since you know that other 68000 machines also have windows and graphic screens, theoretically, you should be able to run your program on any machine with a 68000 processor without too much conversion hassle. Well, yes—if you structure your program properly when writing it.

My rules go like this: Divide your program into three sections: (1) *machine-independent* read-only (application code and constants); (2) *machine-dependent* read-only (interface code and constants); and (3) read-write (uninitialized data).

Section one, containing your application code, will probably be the largest. System calls are not allowed in this section. If you want to make a system call, you must place that system call in a short subroutine in section two, and then call that subroutine from section one. Code and data in section one cannot be changed or written over at run-time.

Section two contains a series of short routines that make the actual calls into the system. Their only function is to insulate the application code in section one from making assumptions about the system. You may find, for example, that sending a character out to the console requires a slightly different calling procedure on the Amiga than on a Mac, an ST, a Sun, or an Apollo. By structuring your program to make system calls only through section two, you can rest assured that this section is the only area of your program that requires modification to perform migration. For disk files, there should ideally be only one subroutine to open a given file, one to close it, one to read a record from it, one to write a record into it, one to create it, one to delete it, and, perhaps, one to append records to it.

Section three contains your scratchpad variables. This section should contain your writable data and your stack. When you start up your program, initialize your stack pointer to the top of your stack +1. You should also be careful not directly address data in this section. Rather, you should place the location of this section in one of the address registers (like A5), and then access the data as an offset from this section's address. This procedure sounds a little strange at first, but it can buy you an incredible bonus: time-sharing. You can have multiple copies of the same program running (such as a bulletin board), using only a *single* copy of sections one and two, and a copy of section three for each user (modem).

This arrangement means you can have a bulletin board program with a 250K section one, a 2K section two, and a 20K section three. If you want to add another user (modem), the 20K for the additional section three is the only additional memory consumed! You can support *many* modems before you run out of RAM because you are running all users from the same copy of the object code.

There are a few small considerations, of course. For instance, there must be a way for each section three to know with which user (modem) he is supposed to be communicating (i.e., a modem number (0..n)). There must also be only one modem character-output routine in section two, and you must pass the modem number to that routine along with the character you want output. Every other routine that interfaces with a modem must also have a modem number passed to it as an argument.

What power! I know of no high-level language that can do that!

In case you are wondering what kind of high-powered, real-time executive is needed to do the switching among the different sections three, it is a surprisingly small one—about one page. I'll show you what it does next month.

Next month, I will cover some of the actual system calls you can use to get characters in and out of the CLI, and, perhaps, some disk I/O calling examples.

•AC•

Roomers

by the Bandito

A special high-resolution graphics board for the Amiga 2000?

From the cover of *Newsweek* to cancellation, in just a few months ... The television series *Max Headroom* has been cancelled, and with it go the use of Amiga graphics on the show. A day after the cancellation, dozens of Amiga systems were removed from the set. As one computer network pundit said, "What? Couldn't they use the Amigas on *Knot's Landing*?"

Commodore has extended the Amiga 1000 to 2000 trade-up offer until the end of November. Reportedly, sales of the 2000 are brisk, and Commodore can't make them fast enough. Meanwhile, rumors continue to surface about problems with the 2000 as developers test fully-tricked-out machines, loaded with expansion cards. Some combinations just don't work, and there has been a lot of finger-pointing between Commodore and hardware developers. Sometimes the finger points at Commodore, sometimes at developers, sometimes between developers.

Baseball hero Earl Weaver was interviewed on the ESPN sports network recently and plugged his game from Electronic Arts, Earl Weaver Baseball. He pulled out a copy of the program and told announcer Roy Firestone that it runs on the Amiga. Thanks, Earl!

Commodore may be planning an expansion chassis for the Amiga 500 that can accept Zorro standard cards. Early guesses say the price will be so high, you may as well buy an Amiga 2000. Hardware developers were quite surprised at this revelation. By and large, Commodore has stayed out of the expansion hardware market.

Along with the rumor of the Commodore expansion box came word that digitizer company New Tek may be working on high-end video graphics hardware. The first sight of it may be at COMDEX in Las Vegas in early November. Meanwhile, video digitizers are coming out of the walls at Amiga shows, and more are said to be on the way from different developers.

Griffin Bacal, Commodore's advertising agency, has hired director Peter Wallach to do the Amiga 500 commercials. Wallach's previous credits include a video for some song called "Thriller."

Beta versions of the new Agnes graphics chip with one megabyte CHIP memory are being circulated to key developers, according to insiders. The latest word says the chips may be at least three months away from distribution.

Commodore may also be preparing a special high-resolution graphics board for the Amiga 2000. It will reportedly have as many as 1024 pixels of horizontal resolution, and perhaps, a four-color video mode at higher resolutions than the Amiga currently boasts. Some estimates say that the black-and-white resolution may be as great at 1280-by-800 pixels. The Workbench software may be limited to a standard 1024 pixels wide, but custom software could access the higher widths.

Commodore hopes to attract the desktop publishing and computer-aided drawing markets with this graphics board. A monitor for these higher resolutions would cost $300–400 for the monochrome display, and about $1500 for color. The odds-on bet is for the monochrome at those resolutions. The price for such a graphics board? About $500. Some developers consider this area to be a blind alley and wish Commodore would work on the Amiga 3000 instead.

Another rumor says Commodore is hot on promoting the Amiga in educational markets with an emphasis on CD ROM technology. The bearer of this rumor says they are trying to compete with a similar campaign being carried out by Apple with the Apple II GS computer. To the Bandito, this makes a certain amount of sense simply because it is so crazy—after all, if Commodore has completely ignored CD ROMs so far, why not declare them to be the latest and greatest thing for the Amiga?

Electronic Arts is dropping the Sierra label from their distribution. Another rumor says Microprose is not interested in doing anymore Amiga stuff and is shifting attention to the Atari ST.

Rumors about the laser toaster keep popping up. Some attendees at the AmiExpo show saw someone walking around with "Laser Toaster Designer" on his show badge. A developer who uses the Amiga to control laser light shows confirmed that lasers can indeed be used to toast bread. The developer said he is thinking of using the toast trick in a rock concert, perhaps for a reunion concert of the group Bread.

•AC•

As I See It

Digi-Paint, Portal, and Videoscape 3D

by Eddie Churchill

Disclaimer: The following article is a composite affair made up of my own opinions, some hard facts, and a rumor or three. I am not ashamed of my opinions. The rest you can take for what it's worth.

Digi-Paint: When is a paint program not a paint program? Every once in a while, a program comes along that I don't think about; I just use it. What I mean is, usually in the course of getting to know a program, I'll say to myself "I wonder how this works," and then proceed to figure it out. Occasionally, though, a program comes along that is sooo tricky that I won't even hazard a guess as to its inner workings. Digi-Paint is such a program. It does so many unbelievable things, it's hard to imagine how it works. The math must be staggering! It makes my head hurt just to think about it. So I don't. I just use it ... and love it.

Digi-Paint does so many things, and all so well; it's hard to know where to start. The shading and tinting must be seen to be believed. HAM brushes shouldn't be possible, if you know anything about Hold And Modify! A pixel's color is determined by the color of the previous pixel. So how do you make a brush that is separate from the main image? Don't ask. You can load two images at once, then cut out a section of one image to see the other image "behind" it. How does Digi-Paint do *that* in HAM? Don't ask. This is what I mean by headache-inducing features. If you try to figure out the magic, you get a headache.

If Digi-Paint has a serious flaw, it's that it's being marketed incorrectly. It is *not* a paint program, at least not in the same sense that Deluxe Paint is a paint program. I suppose if you are ultra-talented, you can sit down and paint with it. That strikes me as overkill, and a clumsy way of accomplishing your end. Digi-Paint is a graphic-arts effects package. You load a picture (either drawn with DP II or digitized with Digi-View) and then "finish" it with Digi-Paint. You can even make a black and white picture look like it was taken in color! This area is where you really use the program to its maximum potential. Which is a lot, because Digi-Paint is complicated and powerful. So don't buy DigiPaint to use it only as a 4096-color painting program—that would be like buying a Cray XMP supercomputer (a cool $4 million) to draw pretty pictures. It can do sooo much more.

Digi-Paint represents the second generation of Amiga software, along with Deluxe Paint II, VideoScape 3D, Word Perfect, Sonix, and Diga!. These are products that are doing more than anyone thought could be done on a personal computer just four years ago. More like them are coming out every week. This is an exciting time to own an Amiga.

Portal: When is a game not a game? This product has been a favorite of mine since I first saw it around last Christmas. The problem is, it's not selling as well as it should. I know why. People don't know what it is. They think it's a game. Nothing could be further from the truth! The box tells it like it is: "A computer novel." That's exactly right. Portal is about as much a game as hiking the Appalachian Trail is a game. Most people would think of a hike from Maine to Georgia (if they thought of such things at all) as a beautiful journey along a predetermined path. *That* is exactly what Portal is: a beautiful journey along a predetermined path. You read it like a book. You don't type anything (It's not a text-adventure.), you don't kill anything (sorry, fantasy role-playing fans), and you don't affect anything along the way (It's not an interactive adventure.).

So it's only a book? What's the big deal about a book? Well, for one, this book comes on three *full* 880K disks. Over 2 and a half meg of book? A typical novel would be about 400K. This book creates an atmosphere of which paper books can only dream. It does this by combining elements we computer users all too often take for granted. Elements like stereo sound, program controlled timing, and text from a computer that seems to be talking to us. By doling out information at its own rate, and having Homer (an AI computer construct who's talking to you) talk about things as if they were real, Portal creates a sense of tension and realism that you've previously had to see a movie to experience. Not bad for "just a book."

A product rarely comes out that is so original and radical that people don't understand it. Portal is such a program. I think Portal deserves a second look by a lot of people. If you love reading Science Fiction as much as I do, you'll like Portal a lot—no matter what you may have heard.

Videoscape 3D:
Animation for the Dedicated

One thing you must say about Aegis: They believe in truth in advertising. Their box for Videoscape 3D is the most truthful thing I've seen in a long time. Two separate items come to mind. The first is a note on the back of the box:

"Note: VideoScape 3D is a sophisticated script-file-based animation tool designed for producing video effects. It is intended for the professional or advanced video hobbyist."

That's no lie! You know how you learn to use most programs like this. You slap it in the drive, boot it up, and start playing around, right? Not this baby. You'll be lost in 15 seconds if you don't read the manual (Which is written very well, by the way. It starts out simple and builds on earlier topics, just like instructional manuals ought to do.).

The second bit of truth is on the side of the box.

"Hardware Options – 1 Megabyte RAM to create animation files."

They don't lie. If you don't have a meg, don't bother trying to record an animation to disk. It's a big no-go. You may as well try to copy a standard workbench disk to ram:. It won't fit.

Not that these are faults! Far from it. I salute Aegis for being honest enough to tell you these things that might limit sales upfront. If you qualify to use VideoScape 3D, you will love it! (If you were frustrated by the Animator, with its lack of real tri-dimensionality, and you have at least 1 meg, you qualify.) This package allows you to get serious about animation. There will be a lot of vertical market sales of the A2000 and VideoScape 3D packaged together.

There are already some spectacular animations hitting the public domain that have to be seen to be believed. They are being broken into pieces for uploading because they are so big, but they *really* demonstrate what a tremendous graphics engine we have here. If you get a chance, download the ShowAnim player and any of the videos you can find on your favorite net or BBS. You won't be sorry. You will probably be surprised.

EndRun

Well, that about wraps up another edition of "As I See It." Before I fade back into the bit-stream, however, I must award the Public Domain program of the month. This month's winner isn't a single program, but the first of a series of lifetime achievement awards to programmers who consistently put up quality goodies. This month, I salute Greg Cunningham. His DirUtil VI or DiskMan 1.3 are utilities that I'm willing to bet the vast majority of Amiga users (at least those who have access to any public domain programs) are using. They are clean, utilitarian, and simple. If you don't have one of his utilities, your life is a lot harder than it has to be. Thanx, Greg. Keep up the good work.

Well, another month has slipped away, and my time is up. Until next month, remember: Piracy—Just Say No!

•AC•

Karate Kid II review
(continued from page 8)

Overview

The game is very fun to play (especially with two people), but does have a few bugs. At times, I've hit an opponent at the right side of the screen, and he's disappeared and reappeared on the left side.

The graphics appear to have been ported from another computer system, but, for the most part, the animation is quite well done. The little men on the screen rub their heads when they fall, and everything on the battle screens moves smoothly.

The sound is good, too. Each time a player kicks or throws a punch, he lets out a yell or grunt. The sound really adds a lot to the game.

This program is copy-protected, and writes scores to the disk. I haven't taken the write protect tab from its "write protected" position, for fear that the scores might ruin the diskette (I've seen this happen with several other games, and I didn't want to take a chance!). Despite the minor problems with this program, I've enjoyed playing it. Karate Kid II isn't the slickest game on the block, but it is still very, very enjoyable.

•AC•

Command Line Arguments in C

by Paul Castonguay

One advantage of the Amiga's disk resident command system is the ease with which existing DOS commands can be modified and new ones added. As a trivial example, MS-DOS users who prefer to use the command name "ERASE" rather than "DELETE" can simply change the name of the erase program in the command directory with this command:

```
Rename Workbench:c/delete as Workbench:c/ERASE
```

Creating new AmigaDOS commands is simple, too. You just add a newly compiled program to the command directory. This makes the program executable at any time, just like any other AmigaDOS command, regardless of where the current directory is or even whether or not the Workbench disk is physically installed in a drive. The Amiga takes care of everything, telling you to install your system disk when it is needed.

Of course, not every program added to the command directory should be considered a new AmigaDOS command. An example is an arcade game like GALAXY-CRUNCH. Why not? Well, I would not view an arcade game as a new command because it is not a useful utility. It does not do something we normally associate with the system commands of a computer (The designer of GALAXY-CRUNCH, however, might legitimately argue with me on that one.).

Would a calculator program like NUMBER-CRUNCH be a valid command? Sure! A calculator that operates right in the CLI window; I'd like that. You might enter NUMBER-CRUNCH 345*894/TAN(43.7)+EXP(28.345) and AmigaDOS would return ... uh ... –1059184.709 (argument of TAN in radians). That might really help an engineering student. And he could say he had added a NUMBER-CRUNCH calculator command to his Amiga.

For this article, I have written a simpler program called Roman which returns the Roman numeral of any Arabic number you enter. Believe it or not, this command exists on the computers made where I work (LISP programming stations for artificial intelligence applications). That is where I got the idea that I should add it as a new command to my Amiga. You simply enter in the CLI window:

```
Roman 29
```

and the Amiga will return:

```
XXIX
```

But wait a minute. Typing "Roman 29" is more complicated than simply typing the name of a program resident in the command directory. Somehow the program must recognize the number 29, or any other number of which I wanted to find the Roman numeral. This is called "passing command line arguments," and this is really the subject of this article.

Passing Command Line Arguments In C

Every time a program is executed from an AmigaDOS CLI window, that program has an opportunity to find out:

1. How many words were entered by the user?
2. What are those words?

This information is passed to the function main() of your program; to get this information, you must use two arguments. You might, in general, write:

```
main(x,y)
```

The information passed would be handed over to the variables x and y. Well, don't forget that you have to properly declare these variables according to some rules of the C language.

First, let's use more informative variable names. Let's declare our main() function of our program like this:

```
main(how_many, where_are_they)
```

I'm taking advantage of the fact that the implementation of Lattice C on the Amiga lets me use variable names which are 30 characters long. (This is an article about the Amiga ... isn't it?) As the name implies, the first argument receives the number of words that have been passed to your program. If you had typed this from the CLI window,

```
Roman 35 42 8636
```

the system (AmigaDOS) would pass the number 4 to the variable "how_many." Did you think it would be 3? The system passes the total number of words the user has entered. The command Roman is itself considered a word. And finally, since the system is passing an integer, I must declare the variable "how_many" as type integer:

```
main(how_many,where_are_they)
int how_many;
```

Now your program can use the variable "how_many" to find out how many arguments were passed to your program. What you do with that information is, of course, up to you. You could have your program respond to only one argument if you wanted. You could have your program issue a warning message if it received the wrong number of arguments. How about: "Enter only one argument you bean-brained human!" Maybe: "Hark! I perceive an over-abundance of formal parameters." Be creative!

Now, let's find out what was passed. You may have heard that C is a language that likes pointers. Pointers are used to find out where things are. In this case, our program knows how many arguments have been passed, but it does not yet know exactly what was passed. In fact, what happens to the words entered on a CLI command line anyway? Surprise! They are conveniently stored in memory for you by AmigaDOS. The location in memory where these arguments are stored is passed to your program via the second variable "where_are_they," as a number representing their address. An address is like a memory location number. We say that the variable "where_are_they" points to the character string the user just entered in the CLI window. You don't have to know exactly what number is. Your program can read it all by itself by looking into the variable "where_are_they."

So, I must declare "where_are_they" as a pointer. Yes, but that's not all. You already know that AmigaDOS allows you to receive more than one argument. Does that mean you need to declare many pointers? One for each argument? Yes and no. Your program does need a separate pointer for each argument passed, but fortunately, you do not have to declare them individually. The system will help you out on this one. All you have to do is declare an array of pointers.

(continued)

An array is a collection of things which all use the same variable name, except that each one is identified by its own particular number, called an index number. The declaration:

```
int bozo[12];
```

This declares an array of integers bozo[0], bozo[1], bozo[2],.....bozo[11]. That's 12 elements starting with 0 and ending with 11. An array of pointers is a series of numbers representing addresses in the computer. They similarly use the same variable name followed by a particular index number. I might declare:

```
char *find_it[20];
```

This would represent 20 places where the addresses of character strings could be stored. The * operator tells the compiler that you have just declared a pointer. I might store the computer address of my favorite girlfriend's name in find_it[19].

```
find_it[19] = &"Matilda";
```

The & is the address operator applied to the string "Matilda." Since I don't know where in the computer my compiler placed the name Matilda, I use the address operator to find out. I could now write:

```
printf("Matilda is in my computer at memory location %u",
find_it[19]);
```

Matilda's location in memory would be reported to the CLI window! Wow! Matilda may not be very impressed but I sure am. Finally, I could write:

```
printf("My favorite gal is %s", find_it[19]);
```

The CLI window would respond by reminding me exactly who it was that I should keep so dear to my heart. Matilda will like that one. The printf command used the address stored in find_it[19] to find and print to screen Matilda's name.

Okay! Okay! Back to passing arguments in C.

I want to declare the variable "where_are_they" as an array of pointers. Here is how it is done:

```
main(how_many, where_are_they)
int how_many;
char *where_are_they[];
```

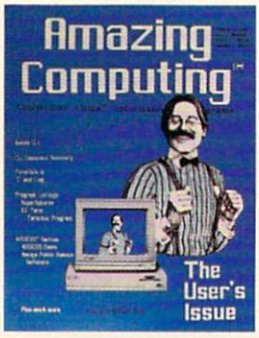
Don't I have to declare the size of the array? No. The array is passed to my program as its beginning address. That's the address of the first pointer variable "where_are_they[0]." The second pointer immediately follows the first. AmigaDOS does that for me. To find the first word passed to my program from the CLI window, I look at the address pointed to by the first element of the pointer array "where_are_they[0]." I could write:

```
printf("The first word is: %s", where_are_they[0]);
```

The word Roman would appear on the screen. Why? Because the first element of the pointer array "where_are_they[0]" points to the first word the user entered in the CLI window. That's the name of the command, Roman. The second element of the pointer array "where_are_they[1]" points to the second word entered by the user. That happens to be the first argument, a number in character format, maybe 29 or 83 or just about anything. The system separates each word by looking for blank spaces. Anything separated by a blank space is treated as, and reported to my program as, a character string address in the pointer array "where_are_they[]." Yes, I did say character string. If you were trying to pass a number like 29, the system would pass you the character string "29." If you wanted to use that number in any calculations, you would first have to convert to the integer 29. I had to do this in my program "Roman," which is Listing Two of this article.

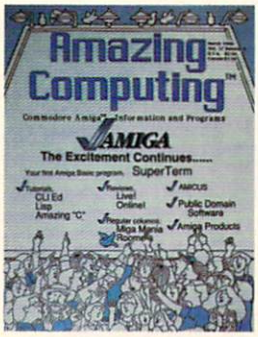
(continued on page 106)

Expanding Reference



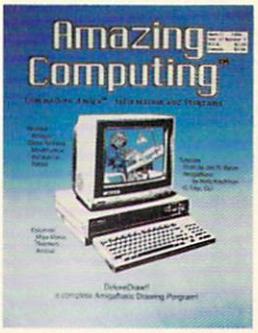
Volume 1 Number 1 Premiere 1986

Super Spheres By Kelly Kauffman An ABasic Graphics prog.
Date Virus By J Foust A disease may attack your Amiga!
EZ-Term by Kelly Kauffman An ABasic Terminal program
Miga Mania by P. Kivolowitz Programming fixes & mouse care
Inside CLI by G. Musser a guided insight into the AmigaDos™
CLI Summary by G. Musser Jr. A list of CLI commands
AmigaForum by B. Lubkin Visit Compuserve's Amiga SIG
Commodore Amiga Development Program by D. Hicks
Amiga Products A listing of present and expected products



Volume 1 Number 2 March 1986

Electronic Arts Comes Through A review of software from EA
Inside CLI: part two G. Musser Investigates CLI & ED
A Summary of ED Commands
Live! by Rich Miner A review of the Beta version of Live!
Online and the CTS Fabite 2424 ADH Modem by J. Foust
Superterm V 1.0 By K. Kauffman A term. prog. in Amiga Basic
A Workbench "More" Program by Rick Wirch
Amiga BBS numbers



Volume 1 Number 3 April 1986

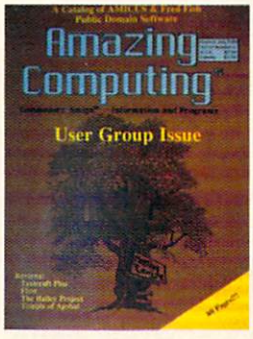
Analyze! a review by Ernest Viveiros
Reviews of Racter, Baratacca and Mindshadow
Forth! The first of our on-going tutorial
Deluxe Draw!! by R. Wirch An Amiga Basic art program
Amiga Basic, A beginners tutorial
Inside CLI: part 3 by George Musser George gives us PIPE

Volume 1 Number 4 May 1986

SkyFox and Arctfox Reviewed
Build your own 5 1/4 Drive Connector By Ernest Viveiros
Amiga Basic Tips by Rich Wirch
Scrimper Part One by P. Kivolowitz prog to print Amiga screen
Microsoft CD ROM Conference by Jim O'Keane
Amiga BBS Numbers

Volume 1 Number 5 1986

The HSI to RGB Conversion Tool
by S. Pietrowicz Color manipulation in BASIC
AmigaNotes by Rick Rae The first of the Amiga music columns
Sidecar A First Look by John Foust A first "under the hood"
John Foust Talks with R. J. Mical at COMDEX™
How does Sidecar affect the Transformer
an interview with Douglas Wyman of Simile
The Commodore Layoffs by J. Foust A look Commodore "cuts"
Scrimper Part Two by Perry Kivolowitz
Marauder reviewed by Rick Wirch
Building Tools by Daniel Kary

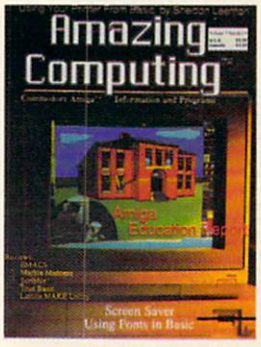


Volume 1 Number 6 1986

Temple of Apshai Trilogy reviewed by Stephen Pietrowicz
The Halley Project: A Mission to our Solar System
reviewed by Stephen Pietrowicz
Flow: reviewed by Erv Bobo
Textcraft Plus a First Look by Joe Lowery
How to start your own Amiga User Group by William Simpson
Amiga User Groups
Mailing List by Kelly Kauffman a basic mail list program
Pointer Image Editor by Stephen Pietrowicz
Scrimper: part three by Perry Kivolowitz
Fun With the Amiga Disk Controller by Thom Sterling
Optimize Your AmigaBasic Programs for Speed by Pietrowicz

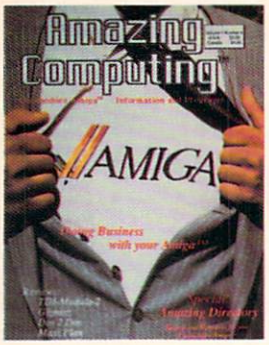
Volume 1 Number 7 1986

Aegis Draw: CAD comes to the Amiga by Kelly Adams
Try 3D by Jim Meadows an introduction to 3D graphics
Aegis Images/ Animator: a review by Erv Bobo
Deluxe Video Construction Set reviewed by Joe Lowery
Window requesters in Amiga Basic by Steve Michel
ROT by Colin French a 3D graphics editor
"I C What I Think" Ron Peterson with a few C graphic progs
Your Menu Sir! by B Catley program Amiga Basic menues
IFF Brush to AmigaBasic 'BOB' Basic editor by M Swinger
Linking C Programs with Assembler Routines on the Amiga
by Gerald Hull



Volume 1 Number 8 1986

The University Amiga By G.Gamble
Amiga's inroads at Washington State University
MicroEd a look at a one man army for the Amiga
MicroEd, The Lewis and Clark Expedition reviewed Frizelle
Scribble Version 2.0 a review
Computers in the Classroom by Robert Frizelle
Two for Study by Frizelle Discovery & TheTalking Coloring Book
True Basic reviewed by Brad Grier
Using your printer with the Amiga
Marble Madness reviewed by Stephen Pietrowicz
Using Fonts from AmigaBasic by Tim Jones
Screen SaVer by P. Kivolowitz A monitor protection prog. in C
Lattice MAKE Utility reviewed by Scott P. Evernden
A Tale of Three EMACS by Steve Poling
.bmap File Reader in Amiga Basic by T Jones



Volume 1 Number 9 1986

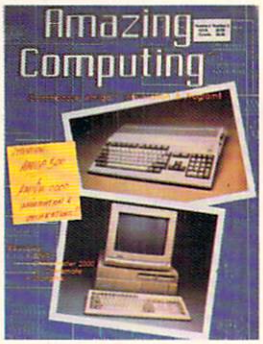
Instant Music Reviewed by Steve Pietrowicz
Mindwalker Reviewed by Richard Knepper
The Alegra Memory Board Reviewed by Rich Wirch
TxEd Reviewed by Jan and Cliff Kent
Amazing Directory A guide to the sources and resources
Amiga Developers A listing of Suppliers and Developers
Public Domain Catalog A listing of Amicus and Fred Fish PDS
Dos 2 Dos review R. Knepper
Transfer files from PC/MS-DOS and AmigaBasic
MaxiPlan review by Richard Knepper The Amiga Spreadsheet
Gizmoz by reviewed by Peter Wayner Amiga extras!
The Loan Information Program by Brian Catley
basic prog. to for your financial options
Starting Your Own Amiga Related Business by W. Simpson
Keep Track of Your Business Usage for Taxes by J. Kummer
The Absoft Amiga Fortran Compiler reviewed by R A. Reale
Using Fonts from AmigaBasic, Part Two by Tim Jones
68000 Macros on the Amiga by G. Hull Advance your ability.
TDI Modla-2 Amiga Compiler review by S Faiwisze

Volume 2 Number 1 1987

What Digi-View is... Or, What Genlock Should Be! by J. Foust
AmigaBasic Default Colors by Bryan Catley
AmigaBasic Titles by Bryan Catley
A Public Domain Modula-2 System reviewed by Warren Block
One Drive Compile by Douglas Lovell
Using Lattice C with a single drive system
A Megabyte Without Megabucks by Chris Irving
An Internal Megabyte upgrade
Digi-View reviewed by Ed Jakober
Defender of the Crown reviewed by Keith Conforti
Leader Board reviewed by Chuck Raudonis
Roundhill Computer System's PANEL reviewed by Ray Lance
Digi-Paint....by New Tek previewed by John Foust
Deluxe Paint II ....from Electronic Arts previewed by J. Foust

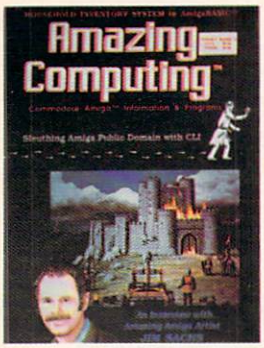
Volume 2 Number 2 1987

The Modem by Josph L. Rothman efforts of a BBS Sysop
MacroModem reviewed by Stephen R. Pietrowicz
GEMINI or "It takes two to Tango" by Jim Meadows
Gaming between machines
BBS-PC! reviewed by Stephen R. Pietrowicz
The Trouble with Xmodem by Joseph L. Rothman
The ACO Project....Graphic Teleconferencing on the Amiga by S. R. Pietrowicz
Flight Simulator II....A Cros Country Tutorial by John Rafferty
A Disk Librarian in AmigaBASIC by John Kennan
Creating and Using Amiga Workbench Icons by C. Hansel
AmigaDOS version 1.2 by Clifford Kent
The Amazing MIDI Interface build your own by Richard Rae
AmigaDOS Operating System Calls and
Disk File Management by D. Haynie
Working with the Workbench by Louis A. Mamakos Prog in C



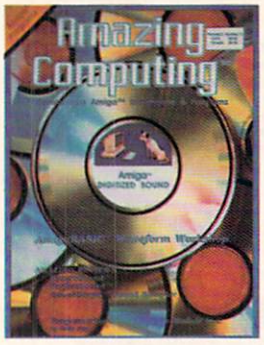
Volume 2 Number 3

The Amiga 2000™ by J Foust
A First look at the new, high end Amiga™
The Amiga 500™ by John Foust
A look at the new, low priced Amiga
An Analysis of the New Amiga PCs by J. Foust
Speculation on the New Amigas
Gemini Part II by Jim Meadows
The concluding article on two-player games
Subscripts and Superscripts in AmigaBASIC by Ivan C. Smith
The Winter Consumer Electronics Show by John Foust
AmigaTrix by W. Block Amiga™ shortcuts
Intuition Gadgets by Harriet Maybeck Tolly
A journey through gadget-land, using C
Shanghai reviewed by Keith M. Conforti
Chessmaster 2000 & Chessmate reviewedby Edwin V. Apel, Jr.
Zing! from Meridian Software reviewed by Ed Bercovitz
Forth! by Jon Bryan Get stereo sound into your Forth programs.
Assembly Language on the Amiga™ by Chris Martin
Roomers by theBandito Genlocks are finally shipping, & MORE!!!
AmigaNotes by R. Rae Hum Busters... "No stereo? Y not?...
The AMICUS Network by J. Foust
CES, user group issues and Amiga Expo*



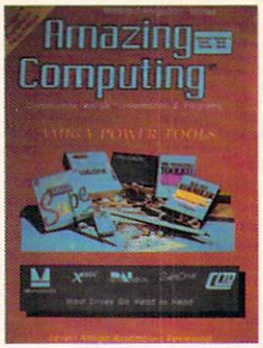
Volume 2 Number 4 1987

Amazing Interviews Jim Sachs by S. Hull Amiga Artist
The Mouse That Got Restored by Jerry Hull and Bob Rhode
Sluething Public Domain Disks with CLI by John Foust
Highlights from the San Francisco Commodore Show by S Hull
Speaker Sessions: San Francisco Commodore Show H Tolly
The Household Inventory System in AmigaBASIC™ by B Catley
Secrets of Screen Dumps by Natkun Okun
Using Function Keys with MicroEmacs by Greg Douglas
Amigatrix II by Warren Block More Amiga shortcuts
Basic Gadgets by Brian Catley Create gadget functions
Gridiron reviewed by K. Conforti Real football for the Amiga
Star Fleet I Version 2.1 reviewed by J. Tracy Amigain Space
The TIC reviewed by J. Foust Battery powered Clock Calendar
Metascope review by H. Tolly An easy-to-use debugger



Volume 2 Number 5 1987

The Perfect Sound Digitizer review by R. Battle
The Future Sound Digitizer by W. Block Applied Vision's SD
Forth! by J. Bryancomparing JForth and Mult-Forth.
Basic Input by B. Catley AmigaBASIC input routine for use in all your programs.
Writing a SoundScape Module in C by T. Fay Programming with MIDI, Amiga and SoundScape by SoundScape author.
Programming in 68000 Assembly Language by C. Martin
Continuing with Counters & Addressing Modes.
Using FutureSound with AmigaBASIC by J. Meadows
AmigaBASIC Programming utility with real, digitzed STEREO
AmigaNotes by R. Rae A review of Mimetics
SoundScape Sound Sampler.
More AmigaNotes by R. Rae
A further review of Sunrize's Perfect Sound .
Waveform Workshop in AmigaBASIC by J. Shields edit & save waveform for use in other AmigaBASIC programs.
The Mimetics Pro MIDI Studio by Sullivan, Jeffery
A review of Mimetics' music editor/player.
Intuition Gadgets Part II by H. MaybeckTolly Boolean gadgets provide the user with an on/off user interface.



Volume 2 Number 6 1987

Forth! by J. Bryan Access resources in the ROM Kernal.
The Amazing Computing Hard Disk Review by J. Foust & S. Leemon In-depth looks at the C Ltd. Hard Drive, Microbotics' MAS-Drive20, Byte by Byte's PAL Jr., Supra's 4x4 Hard Drive and Xebec's 9720H Hard Drive. Also, a look at disk driver software currently under development.

Modula-2 AmigaDOS™ Utilities by S. Faiwiszewsk A Calls to AmigaDOS and the ROM kernal.
Amiga Expansion Peripheral by J. Foust
Explanation of Amiga expansion peripherals.
Amiga Technical Support by J. Foust
How and where to get Amiga tech support.
Goodbye Los Gatos by J. Foust Closing Los Gatos.
The Amicus Network by J. Foust West Coast Computer Faire.
Metacomco Shell and Toolkit by J. Foust A review
The Magic Sac by J. Foust Run Mac programs on your Amiga.
What You Should Know Before Choosing an Amiga 1000 Expansion Device by S. Grant
7 Assemblers for the Amiga by G. Hull Choose your assembler
High Level Shakeup Replaces Top Management at Commodore by S. Hull
Peter J. Baczorby S. Hull Manager at CBM gives an inside look
Logistix A review by Richard Knepper
Organizer by A review Richard Knepper database.
68000 Assembly Language Programming on the Amiga by Chris Martin
Superbase Personal Relational Database by Ray McCabe
AmigaNotes by Rae, Richard A look at FutureSound
Commodore Shows the Amiga 2000 and 500 at the Boston Computer Society by H Maybeck Tolly

Volume 2, Number 7 1987

New Breed of Video Products by John Foust...
Very Vivid! by Tim Grantham...
Video and Your Amiga by Oran Sands III
Amigas & Weather Forecasting by Brenden Larson
A-Squared and the Live! Video Digitizer by John Foust.
Aegis Animator Scripts and Cel Animation by John Foust
Quality Video from a Quality Computer by Oran Sands III.
Is IFF Really a Standard? by John Foust..
Amazing Stories and the Amiga™ by John Foust
All about Printer Drivers by Richard Bielak
Intuition Gadgets by Harriet Maybeck Tolley.
Deluxe Video 1.2 by Bob Eller
Pro Video CG1 by Oran Sands III.
Digi-View 2.0 Digitizer/Software by Jennifer M. Janik
Prism HAM Editor from Impulse by Jennifer M. Janik
Easy! drawing tablet by John Foust..
CSA's Turbo-Amiga Tower by Alfred Aburto
68000 Assembly Language by Chris Martin.

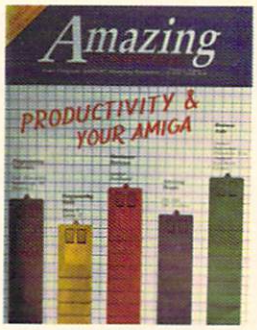
Volume 2, Number 8 1987

This month Amazing Computing™ focuses on entertainment packages for the Amiga. Amazing game reviews...

SDI, Earl Weaver Baseball, Portal, The Surgeon, Little Computer People, Sinbad, StarGlider, King's Quest I, II and III, Faery Tale Adventure, Ultima III, Facets of Adventure, Video Vegas and Bard's Tale.

Plus Amazing monthly columns... Amiga Notes, Roomers, Modula-2, 68000 Assembly Language and The Amicus Network.

Disk-2-Disk by Matthew Leeds
The ColorFonts Standard by John Foust
Skinny C Programs by Robert Riemersma, Jr.
Hidden Messages in Your Amiga™ by John Foust
The Consumer Electronics Show and Comdex by J Foust

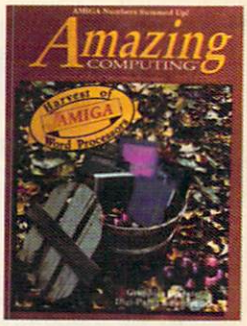


Volume 2 Number 9 1987

Analyze 2.0 reviewed by Kim Schaffer
Impact Business Graphics review by Chuck Raudonis
Microfiche Filer review by Harv Laser
Pagesetter review by Rick Winch
Gizmoz Productivity Set 2.0 review by Bob Eller
Kickwork review by Harv Laser
Diga Telecommunications Package review by Steve Hull
Mouse Time and Timesaver review by John Foust
Insider Memory Expansion review by James O'Keane
Microbotics Starboard-2 review by S. Fawiszewski
Leather Goddess of Phobos
reviewed by Harriet Maybeck-Tolly
Lattice C Compiler Version 3.10 reviewed by Gary Sarff
Manx 3.4a Update reviewed by John Foust
AC-BASIC reviewed by Sheldon Leemon
AC-BASIC Compiler an alternative comparison by B Catley
Modula-2 Programming S Fawiszewski
Raw Console Device Events
Directory Listings Under AmigaDOS by Dave Haynie
AmigaBASIC Patterns by Brian Catley
Programming with Soundscape
Todor Fay manipulate's samples
Bill Volk, Vice-President Aegis Development,
interviewed by Steve Hull
Jim Goodnow, Developer of Manx 'C'
interview by Harriet M Tolly

Volume 2 Number 10 1987

Max Headroom and the Amiga by John Foust
Taking the Perfect Screen Shot by Keith Conforti
Amiga Artist: Brian Williams by John Foust
Amiga Forum on Compuserve™... Software Publishing Conference Transcript by Richard Rae
All About Online Conferencing by Richard Rae
dBMAN reviewed by Clifford Kent
Amiga Pascal reviewed by Michael McNeil
AC-BASIC Compiler reviewed by Bryan Catley
Bug Bytes by John Steiner
Amiga Notes by Richard Rae
Roomers by The Bandito
68000 Assembly Language by Chris Martin
The AMICUS Network by John Foust
Amiga Programming:
Amiga BASIC Structures by Steve Michel
Quick and Dirty Bobs by Michael Swinger
Directory Listings Under Amiga-DOS, Part II by Dave Haynie
Fast File I/O with Modula-2 by Steve Fawiszewski
Window I/O by Read Predmore



Volume 2 Number 11 1987

Word Processors Rundown by Geoff Gamble
ProWrite, Scribble!, and WordPerfect compared
LPD Writer Review by Marion Deland
VizaWrite Review by Harv Laser
Aedit Review by Warren Block
WordPerfect Preview by Harv Laser
Jez San Interview by Ed Bercovitz
The author of StarGlider speaks!
Do-it-yourself Improvements to the Amiga Genlock
Digi-Paint Review by Harv Laser
Sculpt 3D Review by Steve Pietrowicz
Shadowgate Review by Linda Kaplan
TeleGames Review by Michael T. Cabral
Reason Preview: a quick look at an
intense grammar examination application
As I See It by Eddie Churchill Peeking at WordPerfect, Gizmoz V2.0 and Zing! Keys
Bug Bytes by John Steiner
AmigaNotes by R Rae 4 electronic music books
Modula-2 Programming by Steve Fawiszewski
devices, I/O, and the serial port
Roomers by The Bandito
68000 Assembly Language by Chris Martin
Chris walks through the display routines
The AMICUS Network by John Foust
Desktop Publishing & Seybold
C Animation Part II by Mike Swinger Animation Objects
BASIC Text by Brian Catley Pixel perfect text positioning
Soundscape Part III by Todor Fay VU Meter and more
Fun with Amiga Numbers by Alan Barnett
File Browser by Bryan Catley
Full Feature BASIC File Browsing utility

Your Resource to the Commodore Amiga™

The phrase above is not just empty words. The pages of Amazing Computing™ are filled with articles on technical operations and procedures, basic use, and just-plain-fun. The growing library of Amazing Back Issues contains articles from building your own IBM Disk controller, to setting up your own startup sequence. Amazing Computing™ has repeatedly been the first magazine to offer the Amiga users solid, indepth reviews and hands on articles for their machines.

Amazing Computing™ was the first magazine to document CLI
Amazing Computing™ was the first to show Sidecar™ from COMDEX™ in full detail.
Amazing Computing™ was the first to document a 5 1/4 drive connector
Amazing Computing™ was the first with a 1 Meg Amiga upgrade hardware project!
Amazing Computing™ was the first magazine to offer serious programming examples and help.
Amazing Computing™ was the first magazine to offer Public Domain Software at reasonable prices.
Amazing Computing™ was the first magazine with the user in mind!

From the Beginning

Since February 1986, Amazing Computing™ has been providing users with complete information for their Amiga. This store house of programs and information is still available through our back issues. From the Premiere issue to the present, there are insights into the Amiga that any user will find usefull.

$4.00 each!

Our Back Issue price is still $4.00 per issue! (Foreign orders, please add $1.00 per issue for Postage & Handling. All payments must be made by check or money order in U.S. funds drawn on a U.S. Bank.)

Limited Supply

Unforunately, nothing lasts forever, and the availabilty of some of our Back Issues is definitely limited. Please complete your Amazing Computing™ library today, while these issues are still available. Please complete the order form in the rear of this issue and mail with check or money order to:

Back Issues, PiM Publications, Inc., P.O. Box 869, Fall River, MA 02722

(Please allow 4 to 6 weeks for delivery)

Well, there you have it. You now know how to find out how many arguments were typed, and where to find them in memory so you can do something impressive with them. You can pass any argument to a program right on the same line you use to call the program. Install the program in the command directory Workbench:c. Who could argue that you did not add that program as a new command in your Amiga?

Try the example in Listing One first. It will let you try what you have learned. The printf command will cause the number of words entered by the user to be printed on the screen. Also, notice how the value found in the variable "how_many" is used in the condition part of the while loop. It says to print as many words as the system reported entered by the user. The index variable i increments from 0 to the value stored in "how_many" and then causes the loop to end. That's exactly what we want. Type the program using ed (the resident Amiga editor). Compile and link it. I used Lattice C version 3.03. Notice that I did not use any #include "stdio.h." I did not have to. Everything needed to link the printf command to my program is contained in "lstartup.obj" which I must compile with my source code anyway, as per Lattice C's instruction manual. When you finally get your object code, run it from the CLI window. Enter something like this:

<filename> hello out there in the country

Then press return.
Here's what I got:

```
The user has entered 7 words

Word #1 is df1:Amazing_Example1
Word #2 is hello
Word #3 is out
Word #4 is there
Word #5 is in
Word #6 is the
Word #7 is country

That's all folks.
```

Now, take a look at the program Roman.c (Listing Two). This program uses exactly the same method to determine how many arguments have been entered and also to process each one. Naturally, it's a bit more complicated than only printing the arguments to screen. It also has a conditional statement which limits the maximum number of arguments that the user can enter.

Try to apply what you have learned here to applications of your own. How about a program to find the arithmetic mean of a bunch of numbers? Or perhaps a program which returns the size of a file by counting the characters in it? The command could be called HowBig and the filename of the file you want to tally up could be the argument. How about a program which converts Roman numerals back to Arabic?

For now, you can type in the source code, compile it, link it, and install it in your command directory. Tell your friends that you added it yourself as a new command to your Amiga. You will notice numerous attempts to add humor to the program. Try running it with foolish arguments to see the various responses. Try running it to see if it does indeed work correctly. Above all, enjoy!

Listing One

```
    /*
Amazing_Example1.c     */

     main(how_many, where_are_they)

     int how_many;
     char *where_are_they[];

     {
          int i = 0;   /* declare a counter   */

          printf("\nthe user has entered %d words\n\n", how_many);
```

```
        while ( i < how_many )
        {
              printf("Word #%d is %s\n", i + 1, where_are_they[i]);
            i++;
        }
        printf("\nThat's all folks.\n\n");
    }
```

Listing Two

```
/*
Roman.c        */

#include "lattice/math.h"   /*      needed by pow() function        */

main(argc,argv)             /* argc is the number of arguments */
int argc;                   /* argv[] is the pointer array       */
char *argv[];
{
    int length_string = 0;
    int i = 0;
    int arg_number = 1;
    int letter_flag = 0;
    int decimal_point = 0;
    int negative_sign = 0;
    int user_entry = 0;
   double pow();

       while ( arg_number < argc && argc < 10)
     {
          /* How long is the string?  */

         length_string = strlen(argv[arg_number]);

           /*  check for decimal point  */

         decimal_point = 0;
          for (i = 0; i < length_string; i++)
         {
             if (*(argv[arg_number] + i) == 46)
               decimal_point = 1;
         }

           /*    check for negative sign     */

         negative_sign = 0;
          for (i = 0; i < length_string; i++)
         {
            if (*(argv[arg_number]+i) == 45)
              negative_sign = 1;
         }

           /*    check for letters     */

         letter_flag = 0;
         for (i=0; i<length_string; i++)
         {
            if ((*(argv[arg_number]+i)<48 | *(argv[arg_number]+i)>57)
&& (*(argv[arg_number]+i) != 46) && (*(argv[arg_number]+i) != 45))
              letter_flag = 1;
         }
          if ((decimal_point == 1) && (letter_flag == 0))
         {
           printf("\n");
             for (i = 0; i < length_string; i++)
               printf("%c", *(argv[arg_number] + i));
            printf(" is a decimal number.  ");
            printf("The Romans weren't that smart!\n");
           decimal_point = 0;
         }
          else if (letter_flag == 1)
         {
           printf("\n");
             for (i = 0; i < length_string; i++)
               printf("%c", *(argv[arg_number] + i));
            printf(" contains letters.  ");
            printf("Enter only numbers.....you dumbo!!\n");
           letter_flag = 0;
         }
          else if ((negative_sign == 1) && (letter_flag == 0))
         {
           printf("\n");
             for (i = 0; i < length_string; i++)
               printf("%c", *(argv[arg_number] + i));
            printf(" is a negative number.  The Romans didn't know
                    about them.\n");
           negative_sign = 0;
         }
         else
         {
                 /*   Convert to Numeric    */
           user_entry = 0;
            for (i=0; i<length_string; i++)
              user_entry += (*(argv[arg_number]+i)-
48)*pow(10.0,(float)(length_string-1-i));
```

(continued)

```
            if (user_entry > 3999999)
            {
               printf("\n");
                  for (i = 0; i < length_string; i++)
                    printf("%c", *(argv[arg_number] + i));
         printf(" is higher than most Romans knew how to count.\n");
            }
             else if (user_entry == 0)
            {
               printf("\n");
                  for (i = 0; i < length_string; i++)
                    printf("%c", *(argv[arg_number] + i));
                 printf(" did not exist in Caesar's Rome.\n");
            }
            else
            {
               roman(user_entry);
            }
          }
       arg_number++;
       }
    if ( argc > 9)
    {
        printf("\nThat's too much!!  Maybe you'd better buy a bigger
computer.\n");
    }
    if ( argc == 1)
    {
        printf("\nFORMAT: Roman xxx xxx xxx ... where xxx = arabic
number  ...you  bozo!\n");
    }
}
/*
    Find the Roman numeral of any arabic number
    Using Diane's Special algorithm!!!!!                */
roman(arabic)
int  arabic;
{
   char ans1[50],ans2[50];
    int index = 0;
    int question;
    question = arabic;
   arabic=romanize(arabic,1000000,'_','M',ans1,ans2,&index);
    if (arabic >= 900000)
    {
       ans1[index] = '_';
       ans2[index] = 'C';
      index++;
      arabic=romanize(arabic,900000,'_','M',ans1,ans2,&index);
    }
   arabic=romanize(arabic,100000,'_','C',ans1,ans2,&index);
    if (arabic >= 90000)
    {
       ans1[index] = '_';
       ans2[index] = 'X';
      index++;
      arabic=romanize(arabic,90000,'_','C',ans1,ans2,&index);
    }
   arabic=romanize(arabic,10000,'_','X',ans1,ans2,&index);
    if (arabic >= 9000)
    {
       ans1[index] = ' ';
       ans2[index] = 'M';
      index++;
      arabic=romanize(arabic,9000,'_','X',ans1,ans2,&index);
    }
   arabic=romanize(arabic,5000,'_','V',ans1,ans2,&index);
    if (arabic >= 4000)
    {
       ans1[index] = ' ';
       ans2[index] = 'M';
      index++;
      arabic=romanize(arabic,4000,'_','V',ans1,ans2,&index);
    }
    arabic = romanize(arabic, 1000,' ', 'M', ans1, ans2, &index);
    if (arabic >= 900)
    {
       ans1[index] = ' ';
       ans2[index] = 'C';
      index++;
       arabic = romanize(arabic, 900, ' ', 'M', ans1, ans2, &index);
    }
    arabic = romanize(arabic, 500, ' ', 'D', ans1, ans2, &index);
    if (arabic >= 400)
    {
       ans1[index] = ' ';
       ans2[index] = 'C';
      index++;
       arabic = romanize(arabic, 400, ' ', 'D', ans1, ans2, &index);
    }
    arabic = romanize(arabic, 100, ' ', 'C', ans1, ans2, &index);
    if (arabic >= 90)
    {
       ans1[index] = ' ';
       ans2[index] = 'X';
      index++;
       arabic = romanize(arabic, 90, ' ', 'C', ans1, ans2, &index);
    }
    arabic = romanize(arabic, 50, ' ', 'L', ans1, ans2, &index);
    if (arabic >= 40)
    {
       ans1[index] = ' ';
       ans2[index] = 'X';
      index++;
       arabic = romanize(arabic, 40, ' ', 'L', ans1, ans2, &index);
    }
    arabic = romanize(arabic, 10, ' ', 'X', ans1, ans2, &index);
    if (arabic >= 9)
    {
       ans1[index] = ' ';
       ans2[index] = 'I';
      index++;
       arabic = romanize(arabic, 9, ' ', 'X', ans1, ans2, &index);
    }
    arabic = romanize(arabic, 5, ' ', 'V', ans1, ans2, &index);
    if (arabic >=4)
    {
       ans1[index] = ' ';
       ans2[index] = 'I';
      index++;
       arabic = romanize(arabic, 4, ' ', 'V', ans1, ans2, &index);
    }
    romanize(arabic, 1, ' ', 'I', ans1, ans2, &index);
    ans1[index] = '\0';
    ans2[index] = '\0';
    printf("%s\n", ans1);
     printf("%s is the Roman numeral for %d\n", ans2, question);
}
/*            Print the character c as many times as there are
*             j's in the number i, then return i minus the j's.
*/
romanize(i, j, c1, c2, a1, a2, index_address)
int i, j;
char c1, c2, a1[], a2[];
int *index_address;
{
    while(i >= j)
    {
        a1[*index_address] = c1;
        a2[*index_address] = c2;
         *index_address = *index_address + 1;
         i = i - j;
    }
   return(i);
}
```

•AC•

Barry Massoni's

MIDI Interface Adapter

for the Amiga 500 and 2000

The following is one method for converting an Amiga 1000-type MIDI interface to be used with the Amiga 500 or 2000. The conversion is not difficult and does not make the interface useless with the Amiga 1000 because all the modifications are made inside a gender bender. Only 7 components are needed for the conversion which can be done in about an hour.

The serial port gender, pin assignment, and available voltages have been changed on the new Amigas. On the Amiga 500 and 2000, the 5V and –5V power pins from which most interfaces draw their power are not supplied. The conversion reroutes the 12V and –12V power to the correct pins and converts this power from 12V and –12V to the correct 5V and –5V.

First, a few warnings. You must follow the instructions *exactly*. Failure to do so may severely damage the interface or a connected synthesizer. Proceed at your own risk! Either you, I, or the component manufacturers might make a mistake. If you build this device, follow the testing procedures carefully. Testing is the only way to be sure you will not damage something.

If you don't know much about electronics, you will need to know the difference between the anode and cathode of a diode. The band identifies the cathode end (That wasn't so hard ... was it?). Be sure that all diodes are correctly placed.

Here are the assembly instructions. All modifications should be made inside the gender bender. When soldering, always use heat sinks to prevent damage to the components.

1) Designate one side of the gender bender as the Amiga side and one side as the interface side. Mark which side is which. *The interface will not operate if the gender bender is plugged in backwards.*

2) Open the gender bender. Once inside, disconnect all wires from pins 1, 9, and 10 on the Amiga side. You need not remove the wires; just trim off any leads that might create a short circuit.

3) Disconnect all wires from pins 21 and 14 on the MIDI side. Again, you don't have to remove the wires entirely, but rather trim the leads to prevent short circuits.

4) On the Amiga side, connect one 680-ohm resistor to pin 9.

5) Connect the cathode of one of the Zener diodes to the unconnected end of the 680-ohm resistor.

6) Connect the anode of that Zener to pin 1 on the Amiga side.

7) Connect the cathode of the 1N914 diode to pin 21 on the interface side.

8) Connect the anode of the 1N914 to the junction of the Zener and the 680-ohm resistor.

You have now completed a circuit that converts the 12V to 5V and routes the power to the correct pin. Next, you will construct a circuit that converts the –12V to –5V. It is not an exact duplicate of the first circuit, but rather more of a mirror image. Some parts will seem to be backwards in comparison to the first circuit, but this is okay.

9) On the Amiga side, connect the 680-ohm resistor to pin 10.

10) Connect the anode of the 5.1V Zener diode to the unconnected end of the 680-ohm resistor.

11) Connect the cathode of the Zener to pin 1 on the Amiga side. You will notice that it is connected opposite to the first Zener. That's just the way you want it (Yes, both Zeners should be connected to pin 1 on the Amiga side.).

12) Connect the anode of the 1N914 to pin 14 on the interface side. (Again, this is the opposite of the way you connected it on the other circuit.).

13) Connect the cathode of the 1N914 to the junction of the 680-ohm resistor and the Zener anode.

14) Now, very carefully check all the connections that you have made. Look for cold solder joints and reversed diodes. Also make sure there are no short circuits.

15) If you did not understand any of the directions, stop right here and leave me E-mail (Barry Massoni, CIS 73260,1413). I'll try to answer as soon as possible.

16) Turn off your Amiga and connect the interface (still open) to the serial port on your Amiga 500 or 2000.

17) You'll need a voltmeter for this step. I've seen them for less than $10 at hardware stores. If you can't afford one, borrow one. You should not test with your interface and a synth. It's just too expensive if a mistake has been made. Connect the red lead to pin 21 on the interface side and connect the black lead to pin 1 on the Amiga side.

18) Set the meter to read DC at a low range (12 volts or so, but not less than 5 volts).

19) Turn on the Amiga while you are watching the meter. It should almost immediately read 4.5V. If it does not, turn the Amiga off and check all connections again.

20) If it does read correctly, within 1/2V or so, turn off the Amiga.

21) Connect the red lead of the Volt meter to pin 1 on the Amiga side. Connect the black lead to pin 14 on the interface side.

22) Again, turn on the Amiga. As before, it should read within 1/2 volt of 4.5V. If it doesn't read like this, check the connections after turning off the Amiga.

23) If everything reads properly, turn off the Amiga and remove the modified gender bender.

24) Very carefully close the case, making sure no short circuits occur. Be sure no solder joints break when you are closing the case (The results could be disastrous!). Also, make sure the case is non-metallic. If it is metallic, insulate all exposed wires and leads with electrical tape.

25) You've done it! Connect the gender bender to the Amiga and connect the interface to the other end (Be sure you don't have it backwards.). Always make the connections when the Amiga and your synthesizer are both off. Turn everything on and you're ready to go! Have fun!

•AC•

PARTS LIST (Radio Shack Part #)

GENDER BENDER:
This device converts a male DB25 to a female. It is basically just two connected female DB25s. Some are made in cable form; you don't want this type. Try to find the kind made like a small plastic box with a DB25 on each end. The gender bender must be the type you can open—not the permanently-sealed variety.

Two 5.1 VOLT ZENER DIODES (276-565)

Two 1N914 DIODES (276-1620)
(1N4148 diodes may be used instead.)

Two 680 OHM RESISTORS (271-021)
(These are 1/2 watt or 1/4 watt.)
BLUE-GREY-BROWN-SILVER or GOLD)

Jay Miner, the former leader of Amiga Los Gatos and mastermind of the Amiga custom chip set, started things off at the first AmiExpo as the premiere keynote speaker.

by John Foust

The AMICUS Network

The Commodore Show and AmiEXPO

At Amiga shows, you can almost feel the optimism and affection of software developers. It is directed at customers and dealers as usual, but a good share of the devotion is aimed at the Amiga 500. Sales are good, and that means software sales will do well over the next few months. The Amiga 2000 is making developers happy, too. Hardware developers are moving their products to the 2000-sized slots. Software developers are considering applications that take advantage of hard disks and extra memory.

This month's column is a run-down of the new products announced and displayed at two recent Amiga shows, the Commodore Show and AmiEXPO.

With this healthy enthusiasm, it is not surprising to see good attendance at Amiga shows. The Commodore Show was held in Anaheim California at the Disneyland Hotel October 3 and 4. The Commodore 64 will live forever, it seems. The Commodore Show is not Amiga-specific. Vendors showed hardware and software for the 64 and 128. Talking with 64 owners in the crowd, I found that many admired the Amiga, and realized they would need to trade up in the next few months if they wanted to catch the wave.

Early Sunday morning, the Los Angeles area shook from a 5.5 magnitude earthquake, an aftershock from the larger quake several days earlier. The earthquake was the first for many of the vendors who live in other parts of the country. Rumor had it that a certain Amiga hardware company from the Midwest decided not to come to the show because of the recent earthquake.

The First AmiExpo

The first AmiEXPO was held in the Sheraton Centre hotel in Manhattan, New York City. It was the first completely Amiga-specific show of major proportions. It was also the first major show on the East Coast, so I know it attracted Amigaphiles from as far east as Ohio and as far south as Virginia. Being much closer to London, it also attracted many people from Europe.

Each morning opened with a keynote speaker. On Saturday, Jay Miner, the former leader of Amiga Los Gatos and mastermind of the Amiga custom chip set, started things off. Sunday was RJ Mical, the main programmer behind the first version of Intuition. Monday had Richard McIntyre, Commodore's senior vice-president of sales and marketing.

AmiEXPO had as many as four seminars running simultaneously. The staff did an excellent job of arranging speakers. Many of the developers on the show floor were represented in speaking sessions, so if you couldn't talk to them in their booths, you could hear them speak to a crowd. The panels on graphics and video were most heavily attended. I participated on several panels—one on entry-level desktop publishing, one on animation and modeling programs, and one on computer journalism.

Digital Creations, the people who made the Gizmos utilities, revealed several new products at the Commodore Show. They showed a new genlock called Super Gen. The genlock functions can be controlled by software in the Amiga. There is no extra hardware connection to the computer's serial or parallel ports. Instead, they superimpose control information on the video signal using the Amiga custom chips. The video signal is not distorted in any way because the control information is placed in a part of the video signal that is not displayed. The price is $749, and the product will be available in November.

D-Buddy is Digital Creations' new paint program. It edits all resolutions of Amiga images, including HAM, overscan, extra half-bright mode, and pictures larger than the screen. It has some features not found in either Prism or Digi-Paint, the other two HAM paint programs on the market today. It looks good and sells for $79.95.

PAR Software showed Express Paint, a new paint program that supports multiple active brushes, a number of brush special effects, selective backup of regions of the picture, and unlimited use of fonts. Express Paint also supports the extra-half bright mode for a palette of 32 colors and 32 more colors that are half as bright as the first 32 (Remember, not all Amiga 1000s support extra-half bright, but all Amiga 500s and 2000s do.). Express Paint sells for $79.95.

(continued)

Perfect Vision is a new video digitizer from the people who make the Perfect Sound audio digitizer. It digitizes video information with four bits of resolution. With an additional hardware device, it can create color HAM pictures by taking three separate digitizations of the same still video image. It will sell for $219 and is expected to be ready before January. Perfect Vision connects to the parallel port, so it will work on all Amiga models.

At both the Commodore Show and AmiEXPO, A Squared Systems showed the Live! digitizer. It is really, really shipping. No kidding. What more can I say? Mimetics also showed their frame grabber again, which is said to be shipping soon.

Music

Hot Licks is Infinity Software's entry into the music market. An early version of the program was shown at both the Commodore Show and AmiEXPO. Infintiy will sell a plastic overlay keyboard that makes the Amiga keyboard look like a piano (Infinity also showed a game of Go that reportedly plays at the 18 kyu level and can beat the leading Go game on the IBM PC.).

In Anaheim, The Other Guys premiered Synthia, a digital synthesizer, priced at $99.95. It is a system for creating synthesized sounds. Musical sounds can be reproduced and controlled mathematically. Existing or sampled instruments can also be modified with Synthia.

MicroIllusions had a tremendously large booth at the Commodore show. They had thirteen Amigas and a rather complete MIDI instrument setup to demonstrate Music-X, an upcoming MIDI music package.

In New York, Magnetic Music showed the latest incarnation of Texture, a MIDI recording and editing package with special features for developing melodies and rhythm tracks. Sound Quest showed patch librarians for the Yamaha DX-7 and Roland D-50.

Animation

Hash Enterprises, creators of Animator: Apprentice, made their first show appearance at the Commodore Show. The animations produced by their product are first-class. Animator: Apprentice is a full-blown system for creating cartoon-like characters and choreographing them on a stage.

In Anaheim, InnoVision Systems showed a preliminary version of Video Effects 3D, a program that animates IFF brushes in three-dimensions. Given brushes or text in different fonts, and a path of motion through space, Video Effects 3D produces animations that resemble the special effects found on video production systems costing much more than the Amiga. The frames of animation take a minute or two each to produce, but the animation plays back at full video speeds. It will sell for $249 when it becomes available in December.

At both shows, Meridian Software, makers of Zing!, showed the Demonstrator, a program that can record and play back anything on the Amiga. It can record keystrokes and mouse movements within other programs. The Demonstrator demonstration showed the mouse movements of someone drawing in Deluxe Paint, and let the user interact with the demonstration. Speech and text can be added to the recording. The product sells for $39.95.

At AmiExpo, R&DL Productions showed their Light Box software, a program that assists in creating hand-drawn animations. It shows the previous and next frames of the animation sequence. It works in conjunction with a SummaSketch tablet from Summagraphics and should ship soon. R&DL showed a videotape of animation created with this system and animated with their page-flipping routines.

Boing Jackets

If you've wanted one of the shiny black Amiga Boing jackets seen on the backs of former Amiga employees, you can now get one of your own. A company called Boing is taking orders for the jackets at $125 each. You must send a self-addressed, stamped envelope to the address below for an order form.

Ray-tracing Animation

A ray-tracing animation program called Silver made its debut at AmiEXPO. Silver comes from Impulse, the makers of the Prism HAM paint program. Silver lets you build objects from geometric primitives, such as spheres, cubes, cones, and triangles. Each object is assigned motion and color characteristics. The program can render frames of the animation in minutes, building a twenty or thirty-second animation overnight. The results look very much like the Juggler demo, which was done with similar techniques. Impulse promises new features and modules to expand the abilities of Silver, including such things as texture mapping, where the image of an IFF brush is "painted" on the surface of a geometric object in the animation.

At AmiEXPO, Crystal Innovations showed Mouse Trace, a very clever Amiga product. Mouse Trace resembles a drafting aid that holds your mouse in a brace on an extension arm. The brace has a clear crosshair for pinpointing locations on a drawing placed on a drafting table. It works wonderfully for tracing drawings or entering precise information with the mouse. It is a truly original product. It made so much sense to me when I saw it, and it was easy to use. Of course, it works with all existing software. I think this product could substitute for a drawing tablet in many applications. It sells for $59.95.

Gold Disk announced a product called Comic Setter, a program for making comics. It works in sixteen colors and has a library of clip art, as well as the ability to handle both bitmap and line-oriented graphics. It supports all Preferences printers at a user-specified resolution, so 24-pin printers should yield much better output than 9-pin Epson-style printers.

Very Vivid

The Very Vivid people were at AmiEXPO. I think their booth won the "ooo and ahh" award of the show. Their software and hardware system, called Mandala, incorporates a real-time video digitizer with sophisticated software for detecting motion and collision between the digitized images and the graphics on the screen.

The system was up and running in their booth. By standing before the camera and moving your arms in space around you, you interact with the graphics on the screen. One part of the demo was a paint program; another interacted with musical instruments. By touching dots on the screen, you could pluck strings and strike bells. It is very exciting to play with.

I heard about Commodore chairman of the board Irving Gould's visit to the Very Vivid booth. Apparently, he stood by very interested in the interaction between the computer and the person being digitized. Someone

(continued)

suggested he step in front of the camera himself. He demurred and suggested that one of his subordinates step in instead.

Very Vivid also gave a live performance at a local night spot called the Tunnel. Unfortunately, I didn't make it to this presentation. According to one observer, it was a spectacular performance, but not everyone in attendance understood what was happening. The Very Vivid people thought some people obviously mistook it for a choreographed presentation, where the dancer had a premediated series of movements synchronized to the computer images. As soon as the audience realized that the dancer was truly interacting with the computer, there were many more "ooos" and "ahhhs."

Very Vivid planned to give their full stage demonstration at the AmiEXPO show on Monday; unfortunately, arrangements did not work out and the performance was cancelled.

That same night, Amiga artist Sandra Filipucci, whose works have appeared in *Byte* magazine, hosted an event at Mission Graphics Support. The show was titled Digital Monotypes.

C-64 emulators

Software Insight Systems showed the GO-64! emulator for the first time at AmiEXPO (see review in this issue). A friend suggested an idea for an ad for GO-64!—the headline would read "A great step backward."

Another Commodore 64 emulator was shown by Readysoft. It is supposedly very near completion and should be shipping soon. Like GO-64!, it is somewhat slow and not quite a perfect emulation. Until the new one ships, it is difficult to compare the two emulators. The features listed for the ReadySoft emulator look good. The spec sheet says it will support all Amiga disk drives and printers, and that it has a monochrome mode for extra speed.

Games

Discovery Software showed an Amiga version of Arkanoid, a popular arcade game from Taito. Discovery will be producing Amiga versions of other arcade games under an agreement with Taito. MicroSearch, makers of City Desk, showed Head Coach, a football simulation. Psygnosis showed two new games, Barbarian and Terrorpods.

New C compilers

At AmiEXPO, Lattice was selling their C compiler, version 4.0. Software Distillery member John Toebes is manager of the 68000 compiler development at the SAS Institute, the parent company of Lattice.

The new compiler has dozens of exciting new features. Many of the changes are direct results of requests and suggestions from the Amiga community, as well as from programmers within the company. The code size is down and the speed is up. The BLink linker is improved and has become an official Lattice product, so future versions will not be freely distributable. The new compiler more closely meets the proposed ANSI standard for C. Upgrades from version 3.10 are $45 and well worth it. A future article will discuss the features of the new compiler.

Manx Software was also at AmiEXPO, claiming that the new version (3.6) of their Aztec C compiler will be ready soon. The professional and commercial versions of the compiler will include a source level debugger. Manx reports that the next version of the compiler, version 4.2, will be ready early next year and will meet the ANSI recommendations for C.

A source level debugger lets you view the human-readable program source code while the program executes. The debugger has two windows. The top window shows the source code with the next line to be executed highlighted in a different color. The code can be scrolled with an Amiga scroll gadget. The lower area of the window displays debugging information. The source level debugger makes code development much easier. You can check the values of variables without inserting scads of "printf()s" to find out what is happening in the program. You can change the values of variables on the fly. The debugger prints values in the proper form for the variables' types.

ARP

Microsmiths' Charlie Heath sold preliminary versions of the long-awaited ARP library, a collection of replacement AmigaDOS commands for $5 (Microsmiths makes TxEd and FastFonts.). These commands enhance, as well as replace, the old commands. This software will be freely distributable and may make its way onto a future distribution of AmigaDOS. It will also be posted to bulletin boards and a future Fish disk.

The ARP library will remove a layer of the Amiga operating system that has been a thorn in the side of Amiga programmers from the very beginning. Parts of the operating system rely on data in a certain form, endemic to a language called BCPL (It is more popular in Europe than the United States. Its detractors jokingly call it the British Cruddy Programming Language.). BCPL coding conventions make C programming too difficult. The vagaries of BCPL have limited the development of certain programs, such as shells to replace the CLI.

APL

The Spencer Organization showed a version of the APL language. It is fully integrated into the Amiga programming environment (It includes windows and graphics) and sells for $99. It comes with labels with the APL character set for the front of the keys on your keyboard. They have ported this language to most popular microcomputers, so there are bindings to many other third-party software developers for this APL.

Auto Boot

Comspec showed an autobooting hard disk for the Amiga 1000. It is a SCSI 20 megabyte drive that will sell for $1150 if you own a Comspec RAM board, and $1495 otherwise. It uses "black magic" to accomplish hard disk booting within a few seconds of power-up—without a Kickstart disk—according to a company representative with a finger on the Amiga power switch. As he spoke, he flipped the power switch to demonstrate how fast the disk could boot.

A company called Designlab showed a video digitizer box for the Amiga. It worked in black-and-white in 256-by-242 pixels, but future versions will have higher resolutions and color. The box is designed to accept more memory expansion boards to increase the amount of digitized video that can be stored within the box itself. This product is not aimed at the home market, but instead is targeted at video production houses and other artists. Prices start at $2200.

Designlab also announced a print program called FinePrint. FinePrint uses an interesting technique to output high quality black-and-white prints. Using a worn-out dot-matrix printer ribbon, FinePrint builds up a grey-scale image with layers of light ink. It also prints images at any size, from an inch square to hundreds of feet, according to Designlab's promotional literature.

(continued)

On a related subject, Wollner Associates showed the GlennLoc RM-2 sync generator genlock system at the Commodore Show. This is a professional quality genlock that also does fades and keying between video sources and sells for $2995. Again, this product isn't aimed at the home market.

Central Coast Software, the makers of Disk-2-Disk and Dos-2-Dos, announced a WYSIWYG word processor called Precisely and a hard disk backup program called Quarterback that works with all Amiga hard disks.

ASDG

Hardware company ASDG showed preliminary boards for their Satellite Disk Processor, a hard disk controller that promises to be faster than existing Amiga interfaces. The card has its own 68000 processor and 512K of RAM. A socket for a 68881 coprocessor is also provided. The card supports two ST-506 IBM-style hard disks and as many as 56 SCSI devices.

ASDG had promised to show their 2000-and-1 box at the AmiEXPO, but they decided to devote their time and effort to bringing the device into production, instead of completing the prototype to be shown. The 2000-and-1 is an upgrade path for Amiga 1000 owners. It brings much of the expansion ability of the Amiga 2000 into reach of people who don't want to give up their Amiga 1000s. It will accept two Zorro cards of the old style standard, five Zorro cards of the type used in the Amiga 2000, a coprocessor slot, three IBM AT style cards, two 3 1/2 inch drive bays, and a single 5 1/4 inch drive bay.

Byte by Byte showed the Byte Box, a zero to two megabyte FAST memory expansion for the Amiga 500. Byte by Byte is no longer making the PAL Jr. hard disk system, so their attention is now focused on Sculpt 3D, their three-dimensional modeling and ray-tracing program.

Manx Software was among the many exhibitors displaying products at the first AmiExpo in New York City.

Two animations were shown to demonstrate their announced product, Animate 3D, an extension of Sculpt 3D. This new product should be out in late November. One animation showed a rotating Amiga logo passing through a mirrored apple. The other showed one of those executive toys with the bouncing metal balls, sitting on a desktop, complete with books and a desk lamp. Interplay of the lights on the balls and all the shadows were rendered correctly. The frames of the animation were produced with Sculpt 3D and compressed and animated using code from Animate 3D.

Finally Technologies showed an internal 68020 upgrade board for the Amiga 1000 called the Hurricane Accelerator. It runs at 16 megahertz and has a socket for the 68881 floating point coprocessor. An optional board can add up to two megabytes of 32-bit memory. Without the 68020 or 68881 chips, the Hurricane sells for $495. The unpopulated two megabyte memory board sells for the same price. Software runs four to eight times faster with this board installed. A six-hour Sculpt 3D picture can be rendered in about 30 minutes, according to a company representative.

Finally makes several software products as well, including Dr. Xes and Talker. B-Paint is a paint program written completely in AmigaBasic which loads and saves pictures in IFF format. The source code and compiled BASIC versions are included for $39.95. This product would be a great tutorial example for novice AmigaBasic programmers.

At the Commodore Show, Pacific Peripherals showed the SubSystem, an expansion chassis for the Amiga 500 that sits under the computer. It has room for two Zorro style cards and an optional floppy drive and sells for $249.

Amazing Computing

For the first time, PiM Publications had a booth at an Amiga show. The current issue of *Amazing Computing* was given away for a donation to the Amercian Cancer Society.

A number of other Amiga magazines had booths at the show, including two from England, *Enigma* and *Commodore Computing International*. North American Amiga magazines included *AmigaWorld*, *Amiga Sentry*, *RoboCity News*, *Money Machine* and two disk magazines, *AMnews* and *Jumpdisk* .

Don Vandeventer publishes *Money Machine* magazine which focuses on using Commodore computers in business. At these shows, Vandeventer always gives a talk on using your computer as a business tool. His magazine reviews Amiga products and is produced completely with Commodore equipment.

Net party

One night during AmiEXPO, national computer network users from People Link, Compuserve, Delphi, BIX, and Usenet met at a local Irish bar. These parties of network people are as fun as costume parties. Everyone comes in the "costume" of their real-world bodies, which can seem very different from the electronic personalities broadcast on the national compuer networks.

After the party gets rolling, there are always people shaking their heads and smiling at others. These people are perplexed because the friend on the network doesn't quite match the body of the person in real life—especially if the other person looks like a middle-aged suburbanite or a little kid who doesn't shave, and you were expecting the opposite.

Electronic networks are wonderful equalizers. When you read messages from other people on a network, their thoughts and personalities are all that count, not age, race, or gender. As the night drew to a close, there was a toast to the network people who could not be there. At that point, some rushed back home to their modems and screens to check their mail.

At the AmiExpo, WordPerfect Corporation showed their eagerly awaited Amiga version of WordPerfect.

A company called Telegames had a computer set up in the corner of the *Amazing Computing* booth because they needed a location distant from their booth. They make two-player games that are played over a modem link. The games include including chess, checkers, backgammon, and a strategic war simulation.

New AMICUS disk

AMICUS disk 23 is a music disk. The new entry contains twenty-four sampled-sound instruments in the IFF 8SVX format. These are not copyrighted and they are freely distributable. They can be used in Instant Music, Deluxe Music Construction Set, Deluxe Video, SoundScape, and Sonix.

The IFF file format for music is also called SMUS. This disk has a public domain SMUS music player, as well as fourteen public domain songs in SMUS format. There are two programs for converting old music formats to SMUS, one for Music Studio and one for MusicCraft. Once converted to SMUS, these old format songs can be played or modified with most Amiga music programs.

ListINSTR lists the instruments needed to play a Deluxe Music song. A Deluxe Music song file contains a list of the instruments. This program lets you get around the "cannot load all the instruments" message. Also included is a wonderful version of the 1812 Overture for Deluxe Music Construction set, right down to the cannons. Special thanks to Rick Wirch and my local Amiga user group, CAMEO, for producing this disk.

•AC•

(Product Source Listing on Page 120)

Product Source Listing

ASDG, Inc
280 River Road
Suite 54 A
Piscataway, NJ 08854
(201) 540-9670

A-Squared Systems
10 Skyway Lane
Oakland, CA 94619
(415) 633-0703

Boing
1881 Ellwell Drive
Milpitas, CA 95035

Byte By Byte
992 Capital of Texas Highway North
Suite 150
Austin, TX 78759
(512) 343-4357

Central Coast Software
268 Bowie Drive
Los Osos, CA 93402
(805) 528-4906

Comspec
153 Bridgeland Ave, Unit 5
Toronto, Ontario
Canada M6A 2Y6
(416) 787-0617

Crystal Innovations
2286 E. Steel Rd
St. Johns, MI
(517) 224-8683

Designlab
PO Box 419
Oswego, NY 13827

Digital Creations
530 Howe Avenue Suite 208
Sacramento, CA 95825
(916) 344-4825

Discovery Software
262 South 15th Street Suite 300
Philadelphia, PA 19102
(215) 242-4666

Finally Technologies
25 Van Ness, Suite 550
San Francisco, CA 94102
(415) 621-5670

Gold Disk
2179 Dunwin Drive #6
Missisauga, Ontario
Canada L5L 1X2
(416) 828-0913

Hash Enterprises
14201 S.E. 16th Circle
Vancouver, Wash 98684
(206) 256-8567

Impulse, Inc.
6870 Shingle Creek Parkway, Suite 112
Minneapolis, MN 55430
(800) 328-0184

InnoVision Systems
PO Box 743
Hayward, CA 94543
(415) 538-8355

InterActive Softworks
57 Post Street, Suite 811
San Francisco, CA 94104
(415) 956-2660

Lattice Inc
PO Box 3072
Glen Ellyn, IL 60138
(312) 858-7950

Manx Software Systems
PO Box 55
Shrewsbury, NJ 07701
(201) 542-2121

Meridian Software
PO Box 890408
Houston, Texas 77289-0408
(713) 488-2144

Microlllusions
PO Box 3475
Granada Hills, California 91344
(818) 360-3715

Mimetics Corp
PO Box 60238 Station A
Palo Alto, CA 94306
(408) 741-0117

Micro Magic
Suite 320B
261 Hamilton Avenue
Palo Alto, CA 94301
(415) 327-9107

Money Machine
Box 2618
Ocala, FL 32678

PAR Software
PO Box 1089
Elevator Way, Terminal 2
Vancouver, WA 98666
(206) 694-1539

Pacific Peripherals
PO Box 14575
Fremont, CA 94539
(415) 651-1905

R&DL Productions
11-24 46th Avenue
Long Island City, NY 11101
(718) 392-4090

ReadySoft
PO Box 1222
Lewiston NY 14092
(416) 731-4175

Software Terminal
3014 Alta Mere
Fort Worth, TX 76116
(817) 244-4150

Sound Quest
5 Glenaden Avenue East
Toronto, Ontario
Canada M84 2L2
(416) 234-0347

Spencer Organization
366 Kindermack Road
PO Box 248
Westwood, NJ 07675
(201) 666-6011

Software Insight Systems
16 E International Drive
East Granby, CT 06026
(203) 653-4589

SunRize Industries
PO Box 1453
College Station, TX 77841
(409) 846-1311

Supra Corp
1133 Commercial Way
Albany, OR 97321
(503) 967-9075

The Other Guys
55 North Main Street
Suite 301-D
PO Box H
Logan, Utah 84321
(800) 942-9402

Very Vivid
302-1499 Queen Street West
Toronto, Ontario
Canada M6R 1A3
(416) 537-7222

Wollner Associates
3306 Horseman Lane
Falls Church, VA 22042
(703) 533-1236

The AMICUS & Fred Fish Public Domain Software Library

This software is collected from user groups and electronic bulletin boards around the nation. Each Amicus disk is nearly full, and is fully accessible from the Workbench. If source code is provided for any program, then the executable version is also present. This means that you don't need the C compiler to run these programs. An exception is granted for those programs only of use to people who own a C compiler.

The Fred Fish disk are collected by Mr. Fred Fish, a good and active friend of the Amiga.

Note: Each description line below may include something like 'S-O-E-D', which stands for 'source, object file, executable and documentation'. Any combination of these letters indicates what forms of the program are present. Basic programs are presented entirely in source code format.

AMICUS Disk 1

ABasic programs: Graphics

3DSolids	3d solids modeling prog. w/sample data files
Blocks	draws blocks
Cubes	draws cubes
Durer	draws pictures in the style of Durer
FScape	draws fractal landscapes
Hidden	3D drawing program, w/ hidden line removal
JPad	simple paint program
Optical	draw several optical illusions
PaintBox	simple paint program
Shuttle	draws the Shuttle in 3d wireframe
SpaceArt	graphics demo
Speaker	speech utility
Sphere	draws spheres
Spiral	draws color spirals
ThreeDee	3d function plots
Topography	artficial topography
Wheels	draws circle graphics
Xenos	draws fractal planet landscapes

ABasic programs: Tools

AddressBook	simple database program for addresses
CardFile	simple card file database program
Demo	multiwindow demo
KeyCodes	shows keycodes for a key you press
Menu	run many ABasic programs from a menu
MoreColors	way to get more colors on the screen at once, using aliasing
shapes	simple color shape designer Speakit speech and narrator demo

ABasic programs: Games

BrickOut	classic computer brick wall game
Othello	also known as 'go'
Saucer	simple shoot-em-up game
Spelling	simple talking spelling game
ToyBox	selectable graphics demo

ABasic programs: Sounds

Entertainer	plays that tune
HAL9000	pretends it's a real computer
Police	simple police siren sound
SugarPlum	plays "The Dance of the Sugarplum Fairies"

C programs:

ATerm	simple terminal program, S-E
cc	aid to compiling with Lattice C
decvnt	opposite of CONVERT for cross developers
Dotty	source code to the 'dotty' window demo
echox	unix-style filename expansion, partial S,O-D
fasterfp	explains use of fast-floating point math
FixDate	fixes future dates on all files on a disk, S-E
freedraw	simple Workbench drawing prog.,S-E
GfxMem	graphic memory usage indicator, S-E
Grep	searches for a given string in a file with
docs.	ham shows off the hold-and-modify method of color generation
IBM2Amiga	fast parallel cable transfers between an IBM and an Amiga
Mandel	Mandelbrot set program, S-E
moire	patterned graphic demo, S-E
objfix	makes Lattice C object file symbols visible to Wack, S-E
quick	quick sort strings routine
raw	example sample window I/O
setlace	turns on interlace mode, S-E
sparks	qix-type graphic demo, S-E

Other executable programs:

SpeechToy	speech demonstration
WhichFont	displays all available fonts

Texts:

68020	describes 68020 speedup board from CSA
Aliases	explains uses of the ASSIGN command
Bugs	known bug list in Lattice C 3.02
CLICard	reference card for AmigaDOS CLI
CLICommands	guide to using the CLI
Commands	shorter guide to AmigaDOS CLI commands
EdCommands	guide to the ED editor
Filenames	AmigaDOS filename wildcard conventions
HalfBright	explains rare graphics chips that can do more colors
ModemPins	description of the serial port pinout
RAMdisks	tips on setting up your RAM: disk
ROMWack	tips on using ROMWack
Sounds	explanation of Instrument demo sound file format
Speed	refutation of Amiga's CPU and custom chip speed
WackCmds	tips on using Wack

AMICUS Disk 2

C programs:

alib	AmigaDOS object library manager , S-E
ar	text file archive program, S-E
fixobj	auto-chops executable files
shell	simple CLI shell, S-E
sq, usq	file compression programs, S-E
YachtC	a familiar game, S-E
Make	a simple 'make' programming utility, S-E
Emacs	an early version of the Amiga text editor, S-E-D

Assembler programs:

bsearch.asm	binary search code
qsort.asm	Unix compatible qsort() function, source and C test program
setjmp.asm	setjmp() code for Lattice 3.02
SVprintf	Unix system V compatible printf()
trees.o	Unix compatible tree() function, O-D

(This disk formerly had IFF specification files and examples. Since this spec is constantly updated, the IFF spec files have been moved to their own disk in the AMICUS collection.)

John Draper Amiga Tutorials:

Animate	describes animation algorithms
Gadgets	tutorial on gadgets
Menus	learn about Intuition menus

AMICUS Disk 3

C programs:

Xref	a C cross-reference gen., S-E
6bitcolor	extra-half-bright chip gfx demo, S-E
Chop	truncate (chop) files down to size, S-E
Cleanup	removes strange characters from text files
CR2LF	converts carriage returns to line feeds in Amiga files, S-E
Error	adds compile errors to a C file, S
Hello	window ex. from the RKM, S
Kermit	generic Kermit implementation, flakey, no terminal mode, S-E
Scales	sound demo plays scales, S-E
SkewB	Rubik cube demo in hi-res colors, S-E

AmigaBasicProgs(dir)

Automata	cellular automata simulation
CrazyEights	card game
Graph	function graphing programs
WitchingHour	a game

ABasiC programs:

Casino	games of poker, blackjack, dice, and craps
Gomoku	also known as 'othello'
Sabotage	sort of an adventure game

Executable programs:

Disassem	a 68000 disassembler, E-D
DpSide	shows a given set of IFF pictures, E-D
Arrange	a text formatting program, E-D

Assembler programs:

Argoterm	terminal program with speech and Xmodem, S-E

AMICUS Disk 4 Files from the original Amiga Technical BBS

Note that some of these files are old, and refer to older versions of the operating system. These files came from the Sun system that served as Amiga technical support HQ for most of 1985. These files do not carry a warranty, and are for educational purposes only. Of course, that's not to say they don't work.

Complete and nearly up-to-date C source to 'image.ed', an early version of the Icon Editor. This is a little flaky, but compiles and runs.

An Intuition demo, in full C source, including files: demomenu.c, demomenu2.c, demoreq.c, getascii.c, idemo.c, idemo.guide, idemo.make, idemosil.h, nodos.c, and bwrite.c

addmem.c	add external memory to the system
bobtest.c	example of BOB use
consoleIO.c	console IO example
creaport.c	create and delete ports
creastdi.c	create standard I/O requests
creatask.c	creating task examples
diskio.c	example of track read and write
dotty.c	source to the 'dotty window' demo
dualplay.c	dual playfield example
flood.c	flood fill example
freemap.c	old version of 'freemap'
getbobs.c	tools for VSprites and BOBs
gfxmem.c	graphic memory usage indicator
hello.c	window example from RKM
inputdev.c	adding an input handler to the input stream
joystik.c	reading the joystick
keybd.c	direct keyboard reading
layertes.c	layers examples
mousport.c	test mouse port
ownlib.c, ownlib.asm	example of making your own library with Lattice
paratest.c	tests parallel port commands
seritest.c	tests serial port commands
serisamp.c	example of serial port use
prinintr.c	sample printer interface code
prtbase.h	printer device definitions
regintes.c	region test program
setface.c	source to interlace on/off program
setparallel.c	set the attributes of the parallel port
SetSerial.c	set the attributes (parity, data bits, baud rate, etc.) of the serial port
singplay.c	single playfield example
speechtoy.c	source to narrator and phonetics demo
timedely.c	simple timer demo
timer.c	exec support timer functions
timrstuf.c	more exec support timer functions
WhichFont.c	loads and displays all available system fonts

process.i and prtbase.i assmebler include files:

autorqstr.txt	warnings of deadlocks with autorequesters
consoleIO.txt	copy of the RKM console I/O chapter
diskfont.txt	warning of disk font loading bug
fullfunc.txt	list of #defines, macros, functions
inputdev.txt	preliminary copy of the input device chapter
License	information on Workbench distribution license
printer	pre-release copy of the chapter on printer drivers, from RKM 1.1
v11fd.txt	'diff' of .fd file changes from version 1.0 to 1.1
v28v1.diff	'diff' of include file changes from version 28 to 1.0

AMICUS Disk 5 Files from the Amiga Link / Amiga Information Network

Note that some of these files are old, and refer to older versions of the operating system. These files are from Amiga Link. For a time, Commodore supported Amiga Link, aka AIN, for online developer technical support. It was only up and running for several weeks. These files do not carry a warranty, and are for educational purposes only. Of course, that's not to say they don't work.

A demo of Intuition menus called 'menudemo', in C source

wheres.c	find a file searching all subdirectories
bobtest.c	BOB programming example
sweep.c	sound synthesis example

Assembler files:

mydev.asm	sample device driver
mylib.asm	sample library example
mylib.i	
mydev.i	
asmsupp.i	
macros.i	assembler include files

Texts:

amigatricks	tips on CLI commands
extdisk	external disk specification
gameport	game port spec
parallel	parallel port spec
serial	serial port spec
v1.1update	list of new features in version 1.1
v1.1h.txt	'diff' of include file changes from version

Files for building your own printer drivers, including dospecial.c, epsondata.c, init.asm, printer.c, printer.link, printertag.asm, render.c, and wait.asm. This disk does contain a number of files describing the IFF specification. These are not the latest and greatest files, but remain here for historical purposes. They include text files and C source examples. The latest IFF spec is elsewhere in this library.

AMICUS Disk 6 IFF Pictures

This disk includes the DPSide program, which can view a given series of IFF pictures, and the 'showpic' program, which can view each file at the click of an icon. The pictures include a screen from ArtcFox, a Degas dancer, the guys at Electronic Arts, a gorilla, horses, King Tut, a lighthouse, a screen from Marble Madness, the Bugs Bunny Martian, a still from an old movie, the Dire Straits moving company, a screen from Pinball Contruction Set, a TV newcaster, the PaintCan, a world map, a Porsche, a shuttle mission patch, a tyrannosaurus rex, a planet view, a VISA card, and a ten-speed.

AMICUS Disk 7 DigiView HAM demo picture disk

This disk has pictures from the DigiView hold-and-modify video digitzer. It includes the ladies with pencils and lollypops, the young girl, the bulldozer, the horse and buggy, the Byte cover, the dictionary page, the robot and Robert. This includes a program to view each picture separately, and all together as separate, slidable screens. The 'seeibm' program, to turn any screen into an IFF picture.

AMICUS Disk 8

C programs:

Browse	view text files on a disk, using menus S-E-D
Crunch	removes comments and white space from C files, S-E
IconExec	EXECUTE a series of commands from Workbench S-E
PDScreen Dump	dumps Rastport of highest screen to printer
SetAlternate	sets a second image for an icon, when clicked once S-E
SetWindow	makes windows for a CLI program to run under Workbench S-E
SmallClock	a small digital clockin a window menu bar
Scrimper	the screen printer in the fourth AC S-E

Amiga Basic Programs:

(Note: Many of these programs are present on AMICUS Disk 1. Several of these were converted to Amiga Basic, and are included here.)

AddressBook	a simple address book database
Ball	draws a ball
Cload	program to convert Compuserve hex files to binary, S-D
Clue	the game, Intuition driven
ColorArt	art drawing program
DeluxeDraw	the drawing program in the 3rd AC, S-D
Eliza	conversational computer psychologist
Othello	the game, as known as 'go'
RatMaze	3D ratmaze game
ROR	boggling graphics demo
Shuttle	draws 3D pictures of the space shuttle
Spelling	simple spelling program
YoYo	wierd zero-gravity yo-yo demo, tracks yo-yo to the mouse

Executable programs:

3Dcube	Modula-2 demo of a rotating cube
AtIcon	sets a second icon image, displayed when the icon is clicked
AmigaSpell	a slow but simple spell checker, E-D
arc	the ARC file compression progra must-have for telecom, E-D
Bertrand	graphics demo
disksalvage	prog. to rescue trashed disks, E-D
KwikCopy	a quick but nasty disk copy program: ignores errors, E-D
LibDir	lists hunks in an object file E-D
SaveILBM	saves any screen as IFF pic.E-D ??
ScreenDump	shareware screen dump prog, E only
StarTerm	version 2.0, term program, XmodemE-D

Texts:

LatticeMain	tips on fixing _main.c in Lattice
GDiskDrive	make your own 5 1/4 drive
GuruMed	explains the Guru numbers
Lat3.03bugs	bug list of Lattice C version 3.03
MForgeRev	user's view of the MicroForge HD
PrintSpooler	EXECUTE-based print spool prog.

.BMAP files:

These are the necessary links between Amiga Basic and the system libraries. To take advantage of the Amiga's capabilities in Basic, you need these files. BMAPs are included for 'dist', 'console', 'diskfont', 'exec', 'icon', 'intuition', 'layers', 'mathffp', 'mathieeedoubas', 'mathieeesingbas', 'mathtrans', 'potgo', 'timer' and 'translator'.

AMICUS Disk 9

Amiga Basic Programs:

FlightSim	simple flight simulator program
HuePalette	explains Hue, Saturation, & Intensity
Requester	ex. of requesters from Amiga Basic
ScrollDemo	demonstrates scrolling capabilities
Synthesizer	sound program
WorldMap	draws a map of the world

Executable programs:

Boing!	latest Boing! demo,with selectable speed,E
Brush2C	converts an IFF brush to C data instructions, initialization code, E
Brush2Icon	converts IFF brush to an icon, E
Dazzle	graphics demo, tracks to mouse, E
DeciGEL	assembler program for stopping 68010 errors, S-E-D
Klock	menu-bar clock and date display, E
life	the game of life, E
TimeSet	Intuition-based way to set the time & date
EMEmacs	another Emacs, more oriented to word processing, S-E-D
MyCLI	a CLI shell, works without the Workbench, S-E-D

Texts:

FnctnKeys	read function keys from Amiga Basic
HackerSin	explains how to win the game 'hacker'
Ist68010	guide to installing a 68010 in your Amiga
Boing!	latest Boing! demo,with selectablespeed, E
Brush2C	converts an IFF brush to C data instructions, initialization code, E
Brush2Icon	converts IFF brush to an icon, E
Dazzle	graphics demo, tracks to mouse, E
DeciGEL	assembler program for stopping 68010 errors, S-E-D
Klock	menu-bar clock and date display, E
life	the game of life, E
TimeSet	Intuition-based way to set the time date,
EMEmacs	another Emacs, more oriented to word processing, S-E-D
MyCLI	a CLI shell, works without the Workbench, S-E-D

For PDS orders, Please use form on page 128

Texts:

FnctnKeys	explains how to read function keys from Amiga Basic
HackerSin	explains how to win the game 'hacker'
Ist68010	guide to installing a 68010 in your Amiga
PrinterTip	sending escape sequences to your printer
StartupTip	tips on setting up your startup-sequence file
XfmrReview	list of Transformer programs that work

Printer Drivers:

Printer drivers for the Canon PJ-1080A, the C Itoh Prowriter, an improved Epson driver that eliminates streaking, the Epson LQ-800, the Gemini Star-10, the NEC 8025A, the Okidata ML-92, the Panasonic KX-P10xx family, and the Smith-Corona D300, with a document describing the installation process.

AMICUS Disk 10 Instrument sound demos

This is an icon-driven demo, circulated to many dealers. It includes the sounds of an acoustic guitar, an alarm, a banjo, a bass guitar, a bonk, a calliope, a car horn, claves, water drip, electric guitar, a flute, a harp arpegio, a kickdrum, a marimba, a organ minor chord, people talking, pigs, a pipe organ, a Rhodes piano, a saxophone, a sitar, a snare drum, a steel drum, bells, a vibrophone, a violin, a wailing guitar, a horse whinny, and a whistle.

AMICUS Disk 11

C programs

drutil	Intuition-based, CLI replacement manager S-E
cpri	shows and adjusts priority of CLI processes, S-E
ps	shows info on CLI processes, S-E
victex	displays Compuserve RLE pics, S-E

AmigaBasic programs

pointered	pointer and sprite editor program
optimize	optimization ex ample from AC article
calendar	large, animated calendar, diary and date book program
amortize	loan amortizations
brushtoBOB	converts small IFF brushes to AmigaBasic BOB OBJECTS
grids	draw and play waveforms
hilbert	draws Hilbert curves
madlib	mad lib story generator
mailtalk	talking mailing list program
meadows3D	3D graphics program, from A CT™ article
mousetrack	mouse tracking example in hires mode
slot	slot machine game
tctactoe	the game
switch	pachinko-like game
weird	makes strange sounds

Executable programs

cp	unix-like copy command, E
cls	screen clear, S-E
diff	unix-like stream editor uses 'diff' output to fix files
pm	chart recorder performances indicator

Assembler programs

cls	screen clear and CLI arguments example

Modula-2

trails	moving-worm graphics demo
caseconvert	converts Modula-2 keywords to uppercase
Forth	Breshehan circle algorithm example
Analyze	12 templates for the spreadsheet Analyze

There are four programs here that read Commodore 64 picture files. They can translate Koala Pad, Doodle, Print Shop and News Room graphics to IFF format. Getting the files from your C-64 to your Amiga is the hard part.

AMICUS Disk 12

Executable programs

blink	'alink' compatible linker, but faster, E-D
clean	spins the disk for disk cleaners, E-D
epsonset	sends Epson settings to PAR from menu E-D
showbig	view hi-res pics in low-res superbitmap, E-D
speaktime	tell the time, E-D
undelete	undeletes a file, E-D
cnvapidhm	converts Apple][low, medium and high res pictures to IFF, E-D
menued	menu editor produces C code for menus, E-D
quick	quick disk-to-disk nibble copier, E-D
quickEA	copies Electronic Arts disks, removes protection, E-D
bad 1.3	demo of text editor from Microsmiths,E-D

C programs

spin3	rotating blocks graphics demo, S-E-D
popdi	start a new CLI at the press of a button, like Sidekick, S-E-D
vsprite	VSprite example code from Commodore, S-E-D
AmigaBBS	Amiga Basic bulletin board prog., S-D

Assembler programs

star10	makes star fields like Star Trek intro,S-E-D

Pictures

Mount Mandelbrot	3D view of Mandelbrot set
Star Destroyer	hi-res Star Wars starship
Robot	robot arm grabbing a cylinder

Texts

vendors	Amiga vendors, names, addresses
cardco	fixes to early Cardco memory boards
cinclude	cross-reference to C include files
mindwalker	clues to playing the game well
slideshow	make your own slideshows from the Kaleidoscope disk

AMICUS Disk 13

Amiga Basic programs

Routines from Carolyn Scheppner of CBM Tech Support, to read and display IFF pictures from Amiga Basic. With documentation. Also included is a program to do screen prints in AmigaBasic, and the newest BMAP files, with a corrected ConvertFD program. With example pictures, and the SaveILBM screen capture program.

Routines to load and play FutureSound and IFF sound files from Amiga Basic, by John Foust for Applied Visions. With documentation and C and assembler source for writing your own libraries, and interfacing C to assembler in libraries. With example sound.

Executable programs

gravity	Sci Amer Jan 86 gravitation graphic simulation, S-E-D

Texts

MIDI	make your own MIDI instrument interface, with documentation and a hi-res schematic picture.

AMICUS Disk 14

Several programs from Amazing Computing issues:

Tools

Dan Kary's	C structure index program, S-E-D

Amiga Basic programs:

BMAP Reader	by Tim Jones
IFFBrush2BOB	by Mike Swinger
AutoRequester	example
DOSHelper	Windowed help system for CLI commands, S-E-D
PETrans	translates PET ASCII files to ASCII files, S-E-D
C Squared	Graphics program from Scientific American, Sept 86, S-E-D
crlf	adds or removes carriage returns from files, S-E-D
dpdecode	decrypts Deluxe Paint, remo
ves copy	protection, E-D
queryWB	asks Yes or No from the user returns exit code, S-E
vc	VisiCalc type spreadsheet, no mouse control, E-D
view	views text files with window and
slider	gadget, E-D

Qing, Sproing, yaBoing, Zoing are sprite-based Boing! style demos, S-E-D

CLIClock, sClock, wClock are window border clocks, S-E-D

Texts

An article on long-persistance phospor monitors, tips on making brushes of odd shapes in Deluxe Paint, and recommendations on icon interfaces from Commodore-Amiga.

AMICUS Disk 15

The C programs include:

'pr'	a file printing utility, which can print files in the background, and with line numbers and control character filtering.
'fm'	displays a chart of the blocks allocated on a disk.
'Ask'	questions an 'execute' file, returns an error code to control the execution in that batch file
'Stat'	an enhanced version of AmigaDOS 'status' command.
'Dissolve'	random-dot dissolve demo displays IFF picture slowly, dot by dot, in a random fashion.
'PopCLI2'	invoke new CLI window at the press of a key.

The executable programs include:

'Form'	file formatting program through the printer driver to select print styles
'DskCat'	catalogs disks, maintains, sorts,merges lists of disk files
'PSound'	SunRize Industries' sampled sound editor & recorder
'Iconmaker'	makes icons for most programs
'Fractals'	draws great fractal seascapes and mountain scapes.
'3D Breakout'	3D glasses, create breakout in a new dimension
'AmigaMonitor'	displays lists of open files,
memory use,	tasks, devices and ports in use.
'Cosmoroids'	version of 'asteroids' for the Amiga.
'Sizzlers'	high resolution graphics demo written in Modula 2.

Texts:

'ansi.txt'	explains escape sequences the CON: device responds to.
'FKey'	includes template for making paper to sit in the tray at the top of the Amiga keyboard.
'Spawn'	programmer's document from Commodore Amiga, describes ways to use the Amiga's multitasking capabilities in your own programs.

AmigaBasic programs:

'Grids'	draw sound waveforms, and hear them played.
'Light'	a version of the Tron light-cycle video game.
'MigaSol'	a game of solitaire.
'Stats'	program to calculate betting averages
'Money'	"try to grab all the bags of money that you can."

AMICUS 15 also includes two beautiful IFF pictures, of the enemy walkers from the ice planet in Star Wars, and a picture of a cheetah.

AMICUS Disk 16

'juggler'	demo by Eric Graham. a robot juggler bouncing three mirrored balls, with sound effects. Twenty-four frames of HAM animation are flipped quickly to produce this image. You control the speed of the juggling. The author's documentation hints that this program might someday be available as a product.

IFF pictures

parodies of the covers of Amiga World and Amazing Computing magazines.

C programs:

'inputhandler'	example of making an input handler.
'FileZap3'	binary file editing program
'ShowPrint'	displays IFF picture, and prints it.
'Gen'	program indexes and retrieves C structures and variables declared in the Amiga include file system.

Executable Programs:

'FixHunk2'	repairs an executable program file for expanded memory
'ms2smus'	converts Music Studio files to IFF standard 'SMUS' format. I have heard this program might have a few bugs, especially in regards to very long songs, but it works in most cases.
'Missile'	Amiga version of the 'Missile Command' video game,

This disk also contains several files of scenarios for Amiga Flight Simulator II. By putting one of these seven files on a blank disk, and inserting it in the drive after performing a special command in this game, a number of interesting locations are preset into the Flight Simulator program. For example, one scenario places your plane on Alcatraz, while another puts you in Central Park

AMICUS Disk 17

Telecommunications disk whuch contains six terminal programs.

"Comm" V1.33	term prog. with Xmodem, WXmodem,
"ATerm" V7.2	term prog. includes Super Kermit
"VT-100"V2.6	Dave Wecker's VT-100 emulator with Xmodem,Kermit, and scripting
"Amiga Kermit"	V4D(060) port of the Unix C-Kermit
"VTek" V2.3.1	Tektronix graphics terminal emulator based on the VT-100 prog. V2.3 and contains 'latest 'arc' file compression
"AmigaHost"	V0.9 for Compuserve. Includes RLE graphics abilities & CIS-B file transfer protocol.
"FixHunk"	expansion memory necessity
"FixObj"	removes garbage characters from modem received files
"Txt"	filters text files from other systems to be read by the Amiga E.C.
"addmem"	executeable version for use with mem expansion article in AC v2..1
'arc'	file documentation and a basic tutorial on un 'arcing files
"arcre"	for makeing "arc" files E.C.

AMICUS Disk 18

Logo	Amiga version of the popular computer language, with example programs, E-D
TV'Text	Demo version of the TV'Text character generator
PageSetter	Freely distributable versions of the updated PagePrint and PageIFF programs for the PageSetter desktop publishing package.
FullWindow	Resizes any CLI window using only CLI commands, E-D
Life3d	3-D version of Conway's LIFE program, E-D
Defdisk	CLI utility to re-assign a new Workbench disk, S-E-D
Calendar.WKS	Lotus-compatible worksheet that makes calendars
SetKey	Demo of keyboard key re-programmer, with IFF picture to make function key labels, E-D
VPG	Video pattern generator for aligning monitors, E-D
HP-10C	Hewlett-Packard-like calculator, E-D
SetPrefs	Change the Preferences settings on the fly, in C, S-E-D
StarProbe	Program studies stellar evolution. C source included for Amiga and MS-DOS, S-E-D
ROT	C version of Colin French's AmigaBasic ROT program from Amazing Computing. ROT edits and displays polygons to create three dimensional objects. Up to 24 frames of animation can be created and displayed. E-D
Scat	Like lng, windows on screen run away from the mouse, E-D
DK	Decays" the CLI window into dust, in Modula 2, S-E-D
DropShadow2	Adds layered shadows to Workbench windows, E-D

AMICUS Disk 19

This disk carries several programs from Amazing Computing. The IFF pictures on this disk include the Amiga Wake part T-shirt logo, a sixteen-color hi-res image of Andy Griffith, and five Amiga Live! pictures from the Amazing Stories episode that featured the Amiga.

Solve	Linear equation solver in assembly language, S-E-D
Gadgets	Bryan Catley's AmigaBasic tutorial,
Household	Bryan Catley's AmigaBasic household inventory program, S-D
Waveform	Jim Shields' Waveform WorkingBasic, S-D
DiskLib	John Kennan's AmigaBasic disk librarian program, S-D
Subscripts	Ivan Smith's AmigaBasic subscript example, S-D
String, Boolean	C programs and executables for Harriet Maybeck Tolly's Intuition tutorials, S-E-D
Skinny C	Bob Riemersma's example for making small C programs, S-E-D
COMAL.h	Make C look like COMAL header file,
EmacsKey	Makes Emacs function key definitons by Greg Douglas, S-D
AMon 1.1	Snoop on system resource use, E-D
BTE	Bard's Tale character editor, E-D
Size	CLI program shows the size of a given set of files, E-D
WinSize	CLI window utility resizes current window, S-E-D

AMICUS Disk 20

Compactor, Decoder	Steve Michel AmigaBasic tools, S-D
BobEd	BOB and sprite editor written in C,S-E-D
SpriteMasterII	Sprite editor and animator by Brad Kiefer, E-D
BitLab	Bitter chip exploration C program by Tomas Rokicki, S-E-D
FPic	Image processing program by Bob Bush loads and saves IFF images, changes them with several techniques, E-D
Bankn	Complete home banking program, balance your checkbook! E-D
cons	Console device demo program with supporting macro routines.
freemap	Creates a visual diagram of free memory
input.dev	sample input handler, traps key or mouse events
joystick	Shows how to set up the gameportdevice as a joystick.
keyboard	demonstrates direct communications with the keyboard.
layers	Shows use of the layers library
mandelbrot	IFF Mandelbrot program
mouse	hooks up mouse to right joystick port
one.window	console window demo
parallel	Demonstrates access to the parallel port.
printer	opening and using the printer, does a screen dump, not working
print.support	Printer support routines, not working.
proctest	sample process creation code, not working
region	demos split drawing regions
samplefont	sample font with info on creating your own
serial	Demos the serial port
singlePlayfield	Creates 320 x 200 playfield
speechtoy	latest version of cute speech demo
speech.demo	simplified version of speechtoy, with ID requests
text.demo	displays available fonts
timer	demos timer.device use
trackdisk	demos trakcdisk driver

AMICUS Disk 21

Target	Makes each mouse click sound like a gunshot, S-E-D
Sand	Simple game of sand that follows the mouse pointer, E-D
PropGadget	Harriet Maybeck Tolly's proportional gadget example, S-E
EHB	Checks to see if you have extra-half-bright graphics, S-E-D
Piano	Simple piano sound program
CelScripts	Makes cel animation scripts for Aegis Animator, in AmigaBasic

This disk has electronic catalogs for AMICUS disks 1 to 20 and Fish disks 1 to 80. They are viewed with the DiskCat program, included here.

AMICUS Disk 22

Cycles	Light cycle game, E-D
Show_Printll	Views and prints IFF pictures, including larger than screen
PrtDrvGen2.3	Latest version of a printer driver generator
Animations	VideoScape animations of planes and boing ball
Garden	Makes fractal gardenscapes
BasicSorts	Examples of binary search and insertion sort in AmigaBasic

AMICUS Disk 23

An AMICUS disk completely dedicated to music on the Amiga. This disk contains two music players, songs, instruments, and players to bring the thrill of playing "Big Sound" on your Amiga

Instruments	a collection of 25 instruments for playing and creating music. The collection ranges from Cannon to Marimba
List INSTR	program to list the instruments DMCS will not load as well as list the origins for any instrument.
Music	a collecton of 14 Classical pieces
1812Overture	The 16 minute classical feature complete with Cannon!

Three Amiga Music Players:

SMUSPlay
MusicCraft2SMUS
MusicStudio2SMUS

Fred Fish Public Domain Software

Fred Fish Disk 1:

amigademo	Graphical benchmark for comparing amigas.
amigaterm	simple communications program with Xmodem
balls	simulation of the "kinetic thingy" with balls on strings
colorful	Shows off use of hold-and-modify mode.
dhrystone	Dhrystone benchmark program.
dotty	Source to the "dotty window" demo on the Workbench disk.
freedraw	A small "paint" type program with lines, boxes, etc.
gad	John Draper's Gadget tutorial program
gfxmem	Graphical memory usage display prog.
halfbrite	demonstrates "Extra-Half-Brite" mode, if you have it
hello	simple window demo
lattfp	accessing the Motorola Fast Floating Point library from C
palette	Sample prog. to design color palettes.
trackdisk	Demonstrates use of the trackdisk driver.
requesters	John Draper's requester tutorial and example program.
speech	Sample speech demo program. Stripped down "speechtoy".
speechtoy	Another speech demo program.

Fred Fish Disk 2:

alib	Object module librarian.
cc	Unix-like frontend for Lattice C compiler.
dbug	Macro based C debugging package. Machine independent.
make	Subset of Unix make command.
make2	Another make subset command.
microemacs	Small version of emacs editor, with macros, no extensions
portar	Portable file archiver.
xrf	DECUS C cross reference utility.

Fred Fish Disk 3:

gothic	Gothic font banner printer.
roff	A "roff" type text formatter.
ff	A very fast text formatter

For PDS orders, Please use form on page 128

cforth — A highly portable forth implementation. Lots of goodies.
xlisp — Xlisp 1.4, not working correctly.

Fred Fish Disk 4:

banner — Prints horizontal banner
bgrep — A Boyer-Moore grep-like utility
bison — GNU Unix replacement 'yacc', not working.
bm — Another Boyer-Moore grep-like utility
grep — DECUS grep
kermit — simple portable Kermit with no connect mode.
MyCLI — Replacement CLI for the Amiga. V. 1.0
mandel — A Mandelbrot set program, by Robert French and RJ Mical

Fred Fish Disk 5

cons — Console device demo program with supporting macro routines.
freemap — Creates a visual diagram of free memory
input.dev — sample input handler, traps key or mouse events
joystick — Shows how to set up the gameport device as a joystick.
keyboard — demonstrates direct communications with the keyboard.
layers — Shows use of the layers library
mandelbrot — IFF Mandelbrot program
mouse — hooks up mouse to right joystick port
one.window — console window demo
parallel — Demonstrates access to the parallel port.
printer — opening and using the printer, does a screen dump, not working
print.support — Printer support routines, not working.
proctest — sample process creation code, not working
region — demos split drawing regions
samplefont — sample font with info on creating your own
serial — Demos the serial port
singlePlayfield — Creates 320 x 200 playfield
speechtoy — latest version of cute speech demo
speech.demo — simplified version of speechtoy, with IO requests
text.demo — displays available fonts
timer — demos timer.device use
trackdisk — demos trakdisk driver

Fred Fish Disk 6:

compress — like Unix compress, a file squeezer
dadc — analog clock impersonator
microemacs — upgraded version of microemacs from disk 2
mult — removes multiple occuring lines in files
scales — demos using sound and audio functions
setparallel — Allows changing parallel port parameters
setserial — Allows changing serial port parameters.
sortc — quicksort based sort program, in C
stripc — Strips comments and extra whitespace from C source

Fred Fish Disk 7:

This disk contains the executables of the game Hack V 1.0.1.

Fred Fish Disk 8:

This disk contains the C source to Hack on disk 7.

Fred Fish Disk 9:

moire — Draws moire patterns in black and white
MVP-FORTH — Mountain View Press Forth, version 1.00.03A. A shareware version of FORTH from Fantasia Systems.
proff — a more powerful text formatting program
setface — Program to toggle interlace mode on and off.
skewb — a rubic's cube type demo
sparks — moving snake Graphics demo

Fred Fish Disk 10:

conquest — An interstellar adventure simulation game
dehex — convert a hex file to binary
filezap — Patch program for any type of file.
fixobj — Strip garbage off Xmodem transferred files.
iff — Routines to read and write iff format files.
ld — simple directory program
ls — Minimal UNIX ls, with Unix-style wildcarding, in C
sq,usq — file squeeze and unsqueeze
trek73 — Star Trek game
yachtc — Dice game.

Fred Fish Disk 11:

dpslide — slide show program for displaying IFF images with miscellaneous pictures

Fred Fish Disk 12:

amiga3d — Shows a rotating 3 dimensional solid "Amiga sign".
ArgoTerm — a terminal emulator program, written in assembler
arrow3d — Shows a rotating 3 dimensional wire frame arrow.
ld4 — directory listing program
IconExec
SetWindow — two progs for launching progs from Workbench, presently only works under CLI.
SetAlternate — Makes an icon show a second image when clicked once
StarTerm — terminal emulator, with ASCII Xmodem, dialer, more.

Fred Fish Disk 13:

A Bundle of Basic programs, including:

Jpad	toybox	ezspeak
mandlebrot	xmodem	3dsolids
addbook	algebra	ror
amgseq1	amiga-copy	band
bounce	box	brickout
canvas	card$	circle
colorcircles	Copy	cubes1
cutpaste	date	dogstar
dragon	draw	dynamictriangle
Eliza	ezterm	filibuster
fractal	fscape	gomoku	dart
haiku	hal9000	halley	hauntedM
hidden	join	loz	mandel
menu	minipaint	mouse	Othello
patch	pena	pinwheel	gbox
random-circles			
Readme	rgb	rgbtest	Rord
sabotage	salestalk	shades	shapes
shuttle			
sketchpad	spaceart	speakspeach	
speecheasy	spell	sphere	
spiral	striper	superpad	suprshr
talk	terminal		
termtest	tom	topography	triangle
wheels	xenos	xmostriper	

(note: some programs are Abasic, most are Amigabasic, and some programs are presented in both languages)

Fred Fish Disk 14:

amiga3d — update of #12, includes C source to a full hidden surface removal and 3D graphics
beep — Source for a function that generates a beep sound
dex — extracts text from within C source files
dimensions — demonstrates N dimensional graphics
filezap — update of disk 10, a file patch utility
gfxmem — update of disk 1, graphic memory usage indicator
gbmerts IFF brush files to Image struct, in C text.
pdterm — simple ANSI VT100 terminal emulator, in 80 x 25 screen
shell — simple Unix 'csh' style shell
termcap — mostly Unix compatible "termcap" implementation.

Fred Fish Disk 15:

Blobs — graphics demo, like Unix 'worms'
Clock — simple digital clock program for the title bar
Dazzle — An eight-fold symmetry dazzler program. Really pretty!
Fish — double buffered sequence cycle animation of a fish
Monopoly — A really nice monopoly game written in AbasiC.
OkidataDump — Okidata ML92 driver and WorkBench screen dump program.
Polydraw — A drawing program written in AbasiC.
Polyfractals — A fractal program written in AbasiC.

Fred Fish Disk 16:

A complete copy of the latest developer IFF disk

Fred Fish Disk 17:

The NewTek Digi-View video digitizer HAM demo disk

Fred Fish Disk 18:

AmigaDsplay — dumb terminal program with bell, selectable fonts
Ash — Prereiease C Shell-like shell program, history, loops, etc.
Browser — wanders a file tree, displays files, all with the mouse
MC68010 — docs on upgrading your Amiga to use a MC68010
Multidim — rotate an N dimensional cube with a joystick
PigLatin — SAY command that talks in Pig Latin
Scrnprer — Screen image printer
Xlisp1.6 — source, docs, and executable for a Lisp interpreter.

Fred Fish Disk 19:

BlackJack — text-oriented blackjack game
JayMinerSlides — Slides by Jay Miner, Amiga graphics chip designer, showing flowchart of the Amiga internals, in 640 x 400.
Keymap_Test — test program to test the key mapping routines
LockMon — Find unclosed file locks, for programs that don't clean up.

Fred Fish Disk 20:

AmigaToAtari — converts Amiga object code to Atari format
DiskSalv — program to recover files from a trashed AmigaDOS disk.
Hash — example of the AmigaDOS disk hashing function
Hd — Hex dump utility ala Computer Language magazine, April 86
MandelBrots — Mandelbrot contest winners
MultiTasking — Tutorial and examples for Exec level multitasking
Pack — strips whitespace from C source
PortHandler — sample Port-Handler program that performs. Shows BCPL environment clues.
Random — Random number generator in assembly, for C or assembler.
SetMouse2 — sets the mouse port to right or left
SpeechTerm — terminal Emulator with speech capabilites, XModem
TxEd — Demo editor from Microsmith's Charlie Heath

Fred Fish Disk 21

This is a copy of Thomas Wilcox's Mandelbrot Set Explorer disk. Very good!

Fred Fish Disk 22

This disk contains two new "strains" of microemacs.

Lemacs — version 3.6 by Daniel Lawrence. For Unix V7, BSD 4.2, Amiga, MS-DOS, VMS. Uses Amiga function keys, status line, execute, startup files, more.
Pemacs — By Andy Poggio. New features include <ALT> keys as Meta keys, mouse support, higher priority, backup files, word wrap, function keys.

Fred Fish Disk 23

Disk of source for MicroEmacs, several versions for most popular operating systems on micros and mainframes. For people who want to port MicroEmacs to their favorite machine.

Fred Fish Disk 24:

Conques — interstaller adventure simulation game
Csh — update to shell on Disk 14, with built in commands,named variables substitution.
Modula-2 — A pre-release version of the single pass Modula-2 compiler originally developed for MacIntosh at ETHZ. This code was transmitted to the AMIGA and is executed on the AMIGA using a special loader. Binary only.

Fred Fish Disk 25

Graphic Hack — A graphic version of the game on disks 7 and 8. This is the graphics-oriented Hack game by John Toebes. Only the executable is present.

Fred Fish Disk 26

UnHunk — Processes the Amiga "hunk" loadfiles. Collect code, data, and bss hunks together, allows individual specificatio of code, data, and bss origins, and generates binary file with format reminiscent of Unix "a.out" format. The output file can be easily processed by a separate program to produce Motorola "S-records" suitable for downloading to PROM programmer. By Eric Black.
C-kermit — Port of the Kermit file transfer program and server.
Ps — Display and set process priorities
Archx — Yet another program for bundling up text files and mailing or posting them as a single file unit.

Fred Fish Disk 27

ABdemos — Amiga Basic demos from Carolyn Scheppner.
NewConverIFD — creates .bmaps from fd files.
BitPlanes — finds addresses of and writes to bitplanes of the screen's bitmap.
AboutBmaps — A tutorial on creation and use of bmaps.
LoadILBM — loads and displays IFF ILBM pics.
LoadACBM — loads and displays ACBM pics.
ScreenPrint — creates a demo screen and dumps it to a graphic printer.
Dsassem — Simple 68000 disassembler. Reads standard Amiga object files and disassembles the code sections. Data sections are dumped in hex. The actual disassembler routines are set up to be callable from a user program so instructions in memory can be disassembled dynamically. By Bill Rogers.
DvorakKeymap — Example of a keymap structure for the Dvorak keyboard layout. Untested but included because assembly examples are few and far between. By Robert Burns of CA
Hypocycloids — Spirograph, from Feb. 84 Byte.
LinesDemo — Example of proportional gadgets to scroll a SuperBitMap.
MemExpansion — Schematics and directions for building your own homebrew 1 Mb memory expansion, by Michael Fellinger.
SafeMalloc — Program to debug 'malloc()' calls
SciencoDemos — Convert Julian to solar and sidereal time, stellar positions and radial velocity epoch calculations and Gallean satellite plotter. By David Eagle.

Fred Fish Disk 28

ABasic games by David Addison: Backgammon, Cribbage, Milestone, and Othello

Cpp — DECUS 'cpp' C preprocessor, and a modified 'tcc' that knows about the 'cpp', for Manx C.
Shar — Unix-compatible shell archiver, for packing files for travel.
SuperBitMap — Example of using a ScrollLayer, syncing SuperBitMaps for printing, and creating dummy RastPorts.

Fred Fish Disk 29

AegisDraw Demo — Demo program without save and no docs.
Animator Demo — Player for the Aegis Animator files
Cc — Unix-like front-end for Manx C
Enough — Tests for existence of system resources, files, and devices
Rubik — Animated Rubik's cube program
StringLib
Vt100 — VT-100 terminal emulator with Kermit and Xmodem protocols

Fred Fish Disk 30

Several shareware programs. The authors request a donation if you find their program useful, so they can write more software.

BBS — an Amiga Basic BBS by Ewan Grantham
FineArt — Amiga art
FontEditor — edit fonts, by Tim Robinson
MenuEditor — Create menus, save them as C source, by David Pehrson
StarTerm3.0 — Very nice telecommunications by J. Nangano

(Fred Fish Disk#30 is free if requested when ordered with at least three other disks from the collection.)

Fred Fish Disk 31

Life — Life game, uses blitter to do 19.8 generations a second.
Mandelbrot — Version 3.0 of Robert French's program.
MxExample — Mutual exclusion gadget example.
RamSpeed — Measure relative RAM speed, chip and fast.
Set — Replacement for the Manx "set" command for environment variables,with improvements.
Tree — Draws a recursive tree, green leafy type, not files.
TxEd — Crippled demo version of Microsmith's text editor, TxEd.
VDraw — Full-featured drawing program by Stephen Vermeulen.
Xicon — Invokes CLI scripts from icon
Ticon — Displays text files from an icon.

Fred Fish Disk 32

Address — Extended address book written in AmigaBasic.
Calendar — Calendar/diary program written in AmigaBasic.
DosPlus1 — First volume of CLI oriented tools for developers.
DosPlus2 — Second volume of CLI oriented tools for developers.

Executables only:

MacView — Views MacPaint pictures in Amiga low or high res, no sample pictures, by Scott Evernden.
Puzzle — Simulation of puzzle with moving square tiles.
ShowHAM — View HAM pictures from CLI.
Solitaire — ABasiC games of Canfield and Klondike, from David Addison.
Spin3 — Graphics demo of spinning cubes, double-buffered example.
Sword — Sword of Fallen Angel text adventure game written in Amiga Basic.
Trails — Leaves a trail behind mouse, in Modula-2

Fred Fish Disk 33

3dstars — 3d version of the "stars" program below.
Bgmap — Low-level graphics example scrolls bitmap with ScrollVPort.
Dbufgels — Double-buffered animation example for BOBs and VSprites.
DiskMapper — Displays sector allocation of floppy disks.
MemView — View memory in real time, move with joystick.
Oing — Bouncing balls demo
Sproing — Oing, with sound effects.
ScreenDump — Dumps highest screen or window to the printer.
Sdb — Simple database program from a DECUS tape.
Stars — Star field demo, like Star Trek.
TermPlus — Terminal program with capture, library, function keys, Xmodem, CIS-B protocols.
Vt100 — Version 2.0 of Dave Wecker's VT-100 emulator, with scripts & function

Fred Fish Disk 34

Alint — Support files for Gimpel's 'lint' syntax checker
Blink — PD 'slink' compaible linker,faster,better.
Browser — Updated to FF:18 'browser', in Manx, with scroll bars, bug fixes.
Btree — b-tree data structure examples
Btree2 — Another version of 'btree'
Calendar — Appointment calendar with alarm.
Less — File viewer, searching, position by percent, line number.
NewFonts — Set of 28 new Amiga fonts from Bil Fischer
Pr — Background print utility, style options, wildcards.
Requester — Deluxe Paint-type file requester, with sample.

Fred Fish Disk 35

ASendPacket — C example of making asynchronous I/O calls to a DOS handler, written by C-A
ConsoleWindow — C example of getting the Intuition pointer a CON: or RAW: window, for 1.2, by C-A.
DirUtil — Walk the directory tree, do CLI operations from menus
DirUtil2 — Another variant of DirutI.
FileRequester — Lattice C file requester module, with demo driver, from Charlie Heath.
MacView — Views MacPaint pictures in Amiga low or high res, with sample pictures, by Scott Evernden.
Plop — Simple IFF resder program
PopCLI — Sidekick-style program invokes a new CLI, with automatic screen blanking.
QuickCopy — Devenport disk copiers duplicate copy-protected disks.
ScrollPf — Dual playfield example, from C-A, shows 400 x 300 x 2 bit plane playfield on a 320 x 200 x 2 plane deep playfield.
SendPacket — General purpose subroutine to send AmigaDos packets.
SpriteMaker — Sprite editor, can save work as C data structure. Shareware by Ray Larson.
Tracker — Converts any disk into files, for electronic transmission. Preserves entire file structure. Shareware by Brad Wilson.
TriClops 3-D — space invasion game, formerly commercial, now public domain. From Geodesic Publications.
Tsize — Print total size of all files in subdirectories.
Unifdef — C preprocessor to remove given #ifdef'd sections of a file, leaving the rest alone. By Dave Yost
Vtest — VT-100 emulation test program. Requires a Unix system.

Fred Fish Disk 36

Acp — Unix-like 'cp' copy program
Clock — Updated version of clock on disk 15.
Csh — Manx 'csh'-like CLI, history, variables, etc.
DietAid — Diet planning aid organizes recipes, calories
Echo — Improved 'echo' command with color, cursor addressing
FixHunk — Fixs programs to let them run in external memory.
Fm — Maps the sectors a file uses on the disk.
KickBench — Docs, program to make a single disk that works like a Kickstart and Workbench.
Lex — Computes Fog, Flesch, and Kincaid readability of text files.
TunnelVision — David Addison ABasic 3D maze perspective game.
Vc — Visicalc-like spreadsheet calculator program.
Vt100 — Version 2.2 of Dave Wecker's telecom program

YaBoing	Oing! style game program shows sprite collision detects

Fred Fish Disk 37

This disk is a port of Timothy Budd's Little Smalltalk system, done by Bil Kinnersley at Washington State University.

Fred Fish Disk 38

CSquared	Sep 86 Sci American, Circle Squared algorithm
FixObj	Strips garbage off Xmodem transfered object files
Handler	AmigaDOS handler (device) example from C-A
Hp-10c	Mimics a HP-10C calculator, written in Modula-2
IFFEncode	Saves the screen as an IFF file
IffDump	Dumps info about an IFF file
Jsh	BDS C-like CLI shell
NewStat	STATUS-like program, shows priority, processes
Reversi	Game of Reversi, version 6.1
UUdecode	Translate binary files to text, Unix-like programs
Vdraw	Drawing program, version 1.14
VoiceFiler	DX MIDI synthesizer voice filer program
Window	Example of creating a DOS window on a custom screen

Fred Fish Disk 39

AnsiEcho	'echo', 'touch', 'list', 'cls' written in assembler.
Display	Displays HAM images from a ray-tracing program, with example pictures.
Driver	Example device driver source, acts like RAM: disk
Xlisp	XLisp 1.7, executable only

Fred Fish Disk 40

Ahost	Terminal emulator with Xmodem, Kermit and CIS B protocols, function keys, scripts, RLE graphics and conference mode.
AmigaMonitor	Dynamically displays the machine state, such as open files, active tasks, resources, device states, interrupts, libraries, ports, etc.
Arc	Popular file compression system, the standard for transiting files
AreaCode	Program that decodes area codes into state and locality.
Blink	'alink' replacement linker, version 6.5
Cosmo	An 'asteriods' clone.
Dg210	Data General D-210 Terminal emulator
DirUtil	Windowed DOS interface program, V 1.4
DOSHelper	Windowed AmigaDOS CLI help program
PagePrint	Prints text files with headers, page breaks, line numbers
PopCLI	Starts a new CLI with a single keystroke, from any program, With a screen-saver feature. Vesions2, with
SpriteEd	Sprite Editor edits two sprites at a time
X-Spell	Spelling checker allows edits to files

Fred Fish Disk 41

AmigaVenture	Create your own text adventure programs in AmigaBasic.
Csh	Version 2.03 of Dillon's C sh-like shell. Executable only
Dbug	Macro based C debugging package. IFF #2
DualPlayField	example from CBM, update to Intuition manual
GetFile	Heath's file requester, with source
LatXref	Cross reference of Lattice 3.10 header files
Lines	Line drawing demo program
SetFont	Changes font used in a CLI window
Vt100	Version 2.3 of the VT-100 terminal program.

Fred Fish Disk 42

This disk contains an Amiga version of MicroGNUEmacs.

Fred Fish Disk 43

BasicBoing	AmigaBasic program demos page flipping of a 3D cube
Bbm	Demo copy of B.E.S.T. Business Management System.
BbsList	A list of Amiga Bulletin Board Systems
Cc	C compiler frontends for Manix and Lattice C
Copper	A hardware copper list disassembler
InstIFF	Converts Instruments demo sounds to IFF sampled sounds
PopColours	Adjust RGB colors of any screen
SpriteClock	Simple clock is displayed on a sprite above all screens
ST Emulator	Non-serious Atari ST emulator
WBrun	Lets Workbench programs be run from the CLI
Wild	Two Unix shell style wild card matching routines

Fred Fish Disk 44

Icons	Miscellaneous icons
NewIFF	New IFF material from CBM for sampled voice and music files
RayTracePics	The famous ray-tracing pictures, from FF#39, now converted to IFF HAM for "much" faster viewing.
format	
ViewILBM	Displays normal and HAM ILBM files

Fred Fish Disk 45

Clue	Clue board game
Make	Another 'make', with more features
Pictures	Miscellaneous pictures
Update	Updates an older disk with newer files from another disk
Wherels	Searches a disk for files of given name

Fred Fish Disk 46

Asm	Shareware 68010 macro assembler, ROM Kernal Manual compatible
CheckModem	'execute' file program detects presence of modem
Egad	Gadget editor from the Programmers Network
Jive	Transforms a file from English to Jive.
MyLib	A binary only copy of Ma tt's alternate runtime library. Author: Matt Dillon
ProffMacros	Subset Berkeley 'ms' and 'mm' macros for 'proff'
ValSpeak	Transforms a file from English to Valley Speak.

Fred Fish Disk 47

3D-Arm	Simulation of a robotic a rm, very good graphics, teaching tool, including C source.
Juggler	Eric Graham's stunning HAM animation of a robot juggler
VT-100	Version 2.4 of Dave Wecker's terminal emulator, with Xmodem and Kermit file transfer protocols

Fred Fish Disk 48

Bru	Alpha version of a hard disk file archiver
Comm	Version 1.30 of a terminal emulator with phone directories
Csh	Version 2.04 of Matt Dillon's Unix 'csh'-like CLI replacement, including Lattice and Manix C source
Diskperf	Disk benchmark program for Unix and Amiga
Du	Computes disk storage of a file or directory
MemWatch	Program to watch for programs that trash low memory. It attempts to repair the damage, and puts up a requester to inform you of the damage. From the Software Distillery.
Profiler	A realtime execution profiler for Manix C programs. Includes C source.

Fred Fish Disk 49

Cycloids	Update of electronic spirograph from disk 27
DirUtil	Enhanced version of DirUtil from disk 35
MultiDef	Scans a set of object modules and libraries searching for multiply defined symbols
MyUpdate	Disk update utility with options for stripping comments from C header files, and interactive verification of the updating process
Plot	Computes and displays 3 dimensional functions in hires
Polygon	Moire type pattern generator with color cycling
QMouse	Queries whether a mouse button is pressed. This can give a return code that can customize a startup-sequence based on whether a mouse button was pressed.
Touch	Example of setting the datestamp on a file, using a technique from Commodore-Amiga
Trees	More extensive version of the trees program on Disk 31

Fred Fish Disk 50

Asm	Version 1.1 of a shareware 68000 macro assembler, compatible with the Metacomco assembler. This includes an example startup module and more Motorola mnaumonics.
BreakOut	A brick breakout game, uses 3-D glasses
DiskZap	Version 1.1 of a program to edit disks and binary files
FirstSilicon	A smart CLI replacement with full editing and recall of previous commands
Missile	A Missile Command-type game, with sound, in assembler
PerfectSound	Sound editor for a low-cost sound digitzer
Sizzlers	Graphics demos
UnixArc	Version of 'arc' for Unix System V machines, in C
Wombat	Version 3.01 of Dave Warker's terminal emulator

Fred Fish Disk 51

Bison	GNU for Unix 'yacc', working update to disk 4 version
Compress	Update to the file compression program on Disk 6
Cos	"Wheel of Fortune"-type game in AmigaBasic
DiffSed	Unix-like 'diff and 'ssed' for finding the differences between two files, and then recreating the other, given one file, and the list of differences.
Sq, Usq	Portable versions of the CP/M squeeze and unsqueeze

Fred Fish Disk 52

Assign	Replacement for AmigaDOS 'assign' command in C
Fractal	Makes random fractal terrains
Poly, HAMPoly	Workbench-type demos for making polygons in lores and HAM
MxGads	Example of mutual exclusion gadgets with GadgetText
Tek4010	Tektronix 4010 terminal emulator
VDraw	Versions 1.16 and 1.19 of a Deluxe Paint-like drawing program

Fred Fish Disk 53

Animations	Demo animations with player program for Aegis Animator
ARCre	Creates rename scripts for files with long names, so they can be easily 'arc'ed and un'arc'ed.
ARP	Preliminary AmigaDOS replacements for 'break', 'lod', 'chmod', 'echo', 'filenote' and 'makedir'
Compiler	Not fully ported to the Amiga, this is a 68000 C compiler. It wil produce simple assembly language output, but needs a lot of work.
Spreadsheet	Update with source of the 'vc' spreadsheet on disk 36
TarSplit	Port of program to split Unix 'tar'archives
UUencode	Utilities to encode and decode binary files for ASCII transmission, expanding them by 35 percent

Fred Fish Disk 54

Hanoi	Solves Towers of Hanoi Problem in it's own Workbench window. by Ali Ozer
ISpell	Port of a Unix screen oriented, interactive spelling checker. (Expension RAM required) by Pace Willisson
Ing	A Screen of lots of bouncing little windows by Leo 'Bols Ewhac' Schwab
Lav	Displays number of tasks in run queue, averaged over last 1, 5, and 15 minute periods. by Willaim Rucklidge
MIDITools	Programs to play/record through the MIDI I/F. by Fred Cassirer
MoreRows	Program to make the Work Bench Screen larger than normal. by Neil Katin and Jim Mackraz
Tilt	Program to make your Amiga look like it didn't pass vibration testing. by Leo 'Bols Ewhac' Schwab

Fred Fish Disk 55

Csh	V2.05 of Matt Dillon's csh like shell (Modified for Manix C). by Matt Dillon, Modified by Steve Drew
NewStartups	New C Startup modules:
AStartup.asm	with 1.2 fixes and better quote handling.
TWStartup.asm	opens a stdio window, using user specs. by Commodore, posted to BIX by Carolyn Schepper
Palette	Change another program's screen colors. by Carolyn Schepper
PipeDevice	Allows the standard output of one process to be fed to the standard input of another. by Matt Dillon
ScreenSave	Save a normal or HAM mode screen as an IFF file. by Carolyn Schepper
ShanghaiDemo	Demo of the Activision game Shanghai.
SoundExample	A double buffered sound example for Manix C. by Jim Goodnow
Vsprites	A working vsprite example. by Eric Cotton
Vt100	V2.6 of Dave's Vt100 terminal emulator with kermit and xmodem. by Dave Wecker

Fred Fish Disk 56

ClipBoard	Clipboard device interface routines, to provide a standard interface. by Andy Finkle
ConPackets	Demos the use of DOS Packets, ConUnit, etc. by Carolyn Schepper
GetDisks	Program to find all available disk device names and return them as an exec list. by Philip Lindsay
GetVolume	Program to get volume name of the volume that a given file resides on. by Chuck McManis
Icon2C	Reads an icon file and writes out a fragment of C code with the icon data structures. by Carolyn Schepper
MergeMem	Program to merge the MemList entries of sequentially configured RAM boards. by Carolyn Schepper
mCAD	An object oriented drawing program, V1.1 by Tim Mooney

Fred Fish Disk 57

Replaced by FF97 Due to Copyright problems

Fred Fish Disk 58

ASDG-rtd	Extremely useful shareware recoverable ram disk. by Perry Kivolowitz
BigView	Displays any IFF picture, independent of the physical display size, using hardware scroll. by John Hodgson
EGraph	Reads pairs of x and y value from a list of files and draws a formatted graph. by Laurence Turner
HyperBase	Shareware data management system. V1.5
MemClear	Walks through the free memory lists, zeroing free memory along the way. by John Hodgson
NewZAP	A third-generation multi-purpose file sector editing utility. V3.0 by John Hodgeson
RainBow	A Mauraider-Style rainbow generator. by John Hodgson
SMUSPlayers	Two SMUS plays, to play SMUS IFF music formatted files. by John Hodgson
View	A tiny ILBM viewer by John Hodgson
WBdump	JX-80 optimized workbench printer that does not use DumpRPort. by John Hodgson

Fed Fish Disk 59

Browser	Update to browser program on disks 18 and 34. S-E
Browser2	Another different browser program. E
Clock	Clock program with fonts, colors. E
Dme	Dillon text editor V1.22 for programmers. E-D
DropCloth	Puts a pattern on the Workbench backdrop. E-D
DropShadow	Puts shadows on Workbench windows. E-D
FixWB	Similar to DropCloth, but doesn't work yet. S-D
mCAD	Object-oriented drawing program, version 1.2.2. Much improved over disk 56.
Robotroff	Demo of animated pointers on Workbench. S-E-D
Supermort	General compounding/amortization loan calculator. E-D

Fred Fish Disk 60

Various shareware and freeware programs

Blitz	Memory resident file viewer. Very fast. E-D
BlitzFonts	Makes text output faster. E-D
HandShake	Terminal emulator with VT52/VT100/ VT102support. E-D
Med	Mouse-driven text editor version 2.1. E-D
PrDrvGen	Generates printer drivers, version 1.1. Source available from author. E-D
Show	Sideshow-like IFF viewer, V2.1. E-D
Uedit	Customizable text editor V2.0. E-D
Ueturbo	Example Uedit setup macros. S-E-D

Fred Fish Disk 61

ATPatch	Patches Transformer to work under AmigaDOS 1.2. S-E-D
FillDisk	Writes zeroes to free blocks on a disk for security. S-E-D
LPatch	Patch for programs that abort when loading under AmigaDOS 1.2. S-E-D
MicroEmacs	Conroy MicroEmacs V3.8b, newer than disk 22. S-E-D
PearlFont	Like Topaz, but rounded edges.
Terrain	Generates fractal scenery. S-E-D
VSprites	Makes 28 Vsprites, from P&E Book.

Fred Fish Disk 62

This is a port of the Unix game 'Hack', by the Software Distillery, version 1.0.3D.

Fred Fish Disk 63

This is a port of the Unix game 'Larn', by the Software Distillery, version 12.0B.

Fred Fish Disk 64

This is an offical IFF specification disk from Commodore, an update to disk 16.

Fred Fish Disk 65

Awk	Unix text processor, like 'awk'. Doesn't work, but source is included. S-E-D.
MWB	Example of rerouting Workbench window open calls to another custom screen. Version 1.01, S-E-D
CloseWB	Example for closing a custom Workbench screen. S-E-D
Cookie	Generates one-line fortune-cookie aphorisms. S-E-D
JTime	Build-your-own mouse port clock.
MenuBuilder	Creates C source files for menus, based on text descriptions. S-E-D.
NewPackets	CBM tutorial on new packets and structures in AmigaDos 1.2.
PascalToC	Pascal to C translator, not so great. S-E-D
Prep	'ratfor'-like FORTRAN preprocessor. S-E-D
RunBack	Starts programs from CLI, allowing CLI window to close. E-D
SunMouse	This program automatically clicks in windows when the mouse is moved over them. Version 1.0, E-D

Fred Fish Disk 66

AmScsi	Preliminary plans for a SCSI disk controller board.
Asm68k	Macro assembler, version 1.0.1. E-D
Assigned	Example for avoiding DOS insert-disk requester, by scanning the list of 'assign'ed names. S-E-D
Dk	Pretends to eat away at CLI window. S-E-D
Flip	Flips whole screen as a joke. S-E-D
Foogol	Foogol cross-compiler generates VAX assembly code. S-E-D
Free	Prints amount of free space on all drives. S-E-D
MallocTest	malloc/free memory test program. S-E-D
Melt	Pretends to melt the screen. S-E-D
Nart	Graphic flying string demo. S-E-D
Purty	Easy way to set printer attributes from Workbench. E-D
RayTracer	Simple ray tracing program. E-D
SendPackets	Updated CBM examples of packet routines on disk 35. S-E-D
SnapShot	Memory resident screen dump. E-D
TagBBS	Shareware BBS system, version 1.02.

Fred Fish Disk 67

AmCat	Shareware disk cataloging program.
AmigaSpell	Shareware Intuition spelling checker, V2.0. E-D
Bouncer	3-D bouncing ball written in MultiForth, S-E-D
Comm	Terminal program version 1.33, E
Dux5	Another version of DirUtil. S-E-D
HexCalc	Hex, octal, & decimal calculator. E-D
Icons	Various big and siternate image icons.
Mandala	Mandala graphics and sound. E
PersMat	Demo shareware personal file manager.
RSLClock	Menu bar clock version 1.3. E-D
RTCubes	Graphics demo of 3D cubes. E-D
Wheel	"Wheel of Fortune"-type game, in AmigaBasic

Fred Fish Disk 68

This is version MG 1b of the MicroGNUEmacs. Source and executabe areincluded, as well as source for other computers besides the Amiga.

Fred Fish Disk 69

Asm68k	Macro assembler, v1.0.3, E-D
BltLab	Bitter exploring program, in C, S-E-D
Conman	Replacement console device handler adds editing and history to any application that uses CON:, v0.9, E-D
Console	Replacement console routines, in C, S-E-D
Dk	Decays the screen bit by bit, update to disk 66, in Modula-2, S-E-D
Frags	Displays memory fragmentatibe listing the size of free memory blocks, in C, S-E-D
IconType	Change the type of an icon, in C, S-E-D
Make	'make' in Manix C, S-E-D
MonProc	Monitors processes for packet activity, in C,S-ED
MouseClock	Turns mouse pointer into a digital clock,in C,SED
Sb	Browses system structures, from Transactor magazine, v1.0, in C, S-E-D
Spew	Generates 'National Enquirer'-type headlines from rules file. In C,S-E-D
Spool	Three programs to demonstrate multitasking and spooling in a printer spooler. In C, v1.2, S-E-D
Wc	Counts words a la Unix 'wc', but faster, in C,S-ED

Fred Fish Disk 70

This is a disk of shareware programs.

AmigaMonitor	Explores state of the system, v1.13
Arc	Standard file compressor and librarian, v0.23, a port of MS-DOS v5.0. E-D
BlackBook	Phone book program.
DoTil	Intuition-driven file manipulator program,v2.0.
GravityWars	Game of planets, ships and black holes,v1.03.
Jobs	Alternate user interface to CLI and WB, v2.1.

For PDS orders, Please use form on page 128

Lens — Magnifies area around mouse, shows it in a window, v1.0.
Life-3d — 3D version of the classic cellular-automaton game, v1.2.
Logo — Logo language interpreter
SetKey — Demo keymap editor, v1.0
Vpg — Makes displays for aligning video monitors, v1.0.

Fred Fish Disk 71

ArFoil — Makes airfoils using the Joukowski transformation, in C, S-E-D
Amiga Basic — Miscellaneous programs including 3D plot program, a kaleidoscope, C-A logo drawing program file comparison utility string search program, S-E-D
Blocks — A variation of "lines", but with variable color blocks. E-D
Comm — Great terminal program, v1.34, E-D
DiskX — Utility for exploring file system.E-D
Fpic — Simple image processing program that operates on IFF pictures, with several filters, merging images, E-D
IconMk — Makes icons for files , v1.2a, E-D
Icons — New icons
NewFonts — Two new fonts; 'shalt18', an electronic ciruit element font, and 'tbm5', a PC-like font.
PetCLI — An AmigaBASIC CLI shell program.
PWDemo — Demo of the commercial product PowerWindows,v1.2. It aids creation of custom windows, menus, and gadgets, giving C or assembly source. E-D
Rot — Creates and animates 3-D objects, v0.5, E-D
TimeSet — Sets time from Workbench, E-D

Fred Fish Disk 72

This is a disk of IFF pictures.

Fred Fish Disk 73

Add — Customizes existing program menus with Amiga-key shortcuts. Also includes 'until', which waits until a given window is created. Shareware, in C, S-E-D.
AutoIconOpen — Fools WB into thinking mouse has double-clicked icons. In C, S-E-D
Dio — Generic Exec device interface code for opening libraries, getting multiple I/O channels, asynchronous operations, etc. In C, S-E-D.
Dissolve — Slowly displays IFF files, a la Nov 85 Dr. Dobb's program. In C, S-E-D
DTerm — Flexible, reprogrammable terminal program v1.10, E-D
Expose — Re-arranges windows so that at least one pixel of menu bar gadgets are exposed. In C, S-E-D.
Lit — Scans a text file, converts to C-style printable strings.C,v2.0, S-E-D
Lmv — "Long Movie", program views series of IFF picts in quick succession, upto 19 fps. Shareware, E-D
MouseOff — Mouse pointer disappears after ten seconds of non-use. In C, S-E-D
PsrOut — Examples of controlling parallel port with resources instead of the PAR: device.In C,S-E-D
PenPalFont — Child-like font.
RunBackGround — Similar to RunBack on disk 66, runs program from the CLI allowing the CLI window to close. In C,S-E-D
SnapShot — Screendump utility,update FF 66.E-D
TypeAndTell — Example installs a device handler before Intuition, and speaks each key as it is pressed. In C and assembler, S-E-D
Xplor — Prints info about system lists, in assembler,S-E-D

Fred Fish Disk 74

Ced — Edits and recalls CLI commands, v1.3, E-D
Control — Intercepts graphic printer dump calls and accesses color map, width,and screen resolution. C,S-E-D
Dme — Simple WYSIWYG text editor for programmers,v1.25. Update of FF 59.E-D
DropShadow — Workbench dropshadows, v2.0. Update to disk 59. E-D
Funds — AmigaBASIC program tracks mutual or stock p-D
Less — Text viewing program, like Unix "more", v1.1, update to disk 34. S-E-D
Makemake — Scans C source files and constructs a vanilla 'makefile' in the current directory. S-E-D
mCAD — Object-oriented drawing prog. v1.2.4, update to FF 59.Shareware, E-D
Random — Simple random number generator in C. S-E-D
TDebug — Monitors devices by intercepting Exec SendIO() and DoIO() vectors, in C, v1.0, S-E-D
Units — Converts measurements in different units, includes "chart" option, in C, S-E-D
XCopy — Replacement for AmigaDOS 'copy', doesn't change the date, uses Unix wildcards. E-D

Fred Fish Disk 75

Bezier — Play with Bezier curves points and granularity, S-E-D
BSplines — Play with b-splines, as above, S-E-D
Comm — C source for Comm terminal program v1.34. S-E-D
Copy — Replacement 'copy' command v1.0, preserves date, in C, S-E-D
Diff — Simple 'diff' in C, S-E-D
DuM2 — Another DirUtil in Modula-2, v1.5, S-E-D
Eless — Fast 'dir' program in C, S-E-D
Fd — Faster 'eless' in C, S-E-D
HardCopy — Sends a transcript of a CLI session to a file, in C, S-E-D
MouseOff — Update to disk 73, turns off mouse pointer, S-E-D
SetFont — Changes the font in a Workbench screen, v2.0, S-E-D
SpeedDir — Another fast 'dir', in assembler, S-E-D

Fred Fish Disk 76 & 77

These are disks 1 and 2 of Chris Gray's Draco distribution for the Amiga. Draco is a compiled, structured language reminiscent of both C and Pascal. A full interface to AmigaDOS and Intuition is supplied. Be sure to get both disk 76 and 77.

Fred Fish Disk 78

Cycles — Cycle game like "Tron", v1.0, E-D
EOMS — Experts Only Mercenary Simulator game, E-D
MandelVroom — Mandelbrot generator with enhanced palette controls, fixed/floating point, presets, v1.50, in Manx C, S-E-D

Fred Fish Disk 79

AsmTools — CLI tools in assembler: echo, loadit, mounted, setlace, why; S-E-D
AssignDev — Give devices multiple names, in C, S-E-D
AuxHandler — Example of a dos handler that allows use of a CLI via the serial port. Includes source. Author: Steve Drew
Cmd — Redirects printer output to a file, in C, S-E-D
Info — AmigaDOS 'info' replacement, in C and assembler, S-E-D
Kill — Removes a task and its resources, in C, S-E-D
M2Error — Displays errors from TDI Modula-2 compiles, S-E-D
MonProc — Update to process packet program from disk 69, in C, S-E-D
Mounted — Program for testing if a drive is present, in a script In C, S-E-D
Nro — Another 'roff'-style text formatter, in C, S-E-D
ParTask — Finds parent task, in C, S-E-D
QueryAny — For scripts, asks a question, accepts Y/N, gives return code. In assembler, S-E-D
ScnSizer — Resets preferences settings for screen size, in C, S-E-D
SharedLib — Example of a shared library, in C and assembler, S-E-D
Task — Simple CreateTask() example in C, S-E-D
Uw — Unix Windows client v1.0, in C, S-E-D
Who — Lists tasks on ready and wait queues, in C, S-E-D

Fred Fish Disk 80 (see Fred Fish 90)

Fred Fish 80 has been withdrawn due to copyright problems.

Fred Fish Disk 81

Asm68k — V1.1.0 of a macro assembler
AutoFacc — Shrinks the FACC window and moves it to the back
Brushes — 53 custom IFF brushes of electronic symbols
CheckIFF — Checks structure of an IFF file CiedV1.4 update to disk 74 of a simple command line editor
Conman — Replaces console handler to add editing and history to many programs
Fonts — Miscellaneous fonts
Icon — V5.0 of the Icon programming language
KeyLock — Freezes the keyboard and mouse until pass word entered.
ScstDisplay — hack created from "Ing"
Smush — Smushes an IFF file.
Target — Each mouse clikc becomes a gunshot

Fred Fish Disk 82

AdventureA port of the classic Crowther and Woods Adventure game
AmicTerm — V0.50 of a telecommunications program, with scripts, redial, beeps, enhanced file requester
D2D-Demo — Demo version of Disk-2-Disk from Central Coast Software
DX-Synth — Voice filer program for Yamaha DX series synthesizers, update to disk 38
DiskMan — V1.0 of another DirUtil program
Icons — Miscellaneous new icons
Panl — Universal MIDI patch panel, v1.2
Rocket — Another Workbench hack, plays Lunar Lander
Sand — Game of sands following your pointer.

Fred Fish Disk 83

This disk contains a demo version of TeX from N Squared. It is limited to small files, and the previewer can only display ten pages or less, and only a small number of fonts are provided.

Fred Fish Disk 84

AudioToolsPrograms from Rob Peck's July/August Amiga World article
BitLab — Bitter experimentation program, V1.2, update to disk 69
Ed — Simple editor, similar to Unix 'ed', based on the editor in Software Tools.
GravityWars — Game of planets, ships and black holes, v1.04, update to disk 70.
HunkPad — Adds legal padding to executables for Xmodem transmission.
PipeHandler — An AmigaDOS pipe device which supports named pipes and taps. V1.2
PopCLI — V3.0 of a hot-key to invoke a CLI window, with screen blanker, update to disk 40.
Requester — Update to disk 34 of a file requester similar to DPaint.
ScsiDevice — V33.1 of a 'mountable MicroForge SCSI driver.
Vtacom — Another Schwab hack, makes TV-like static on screen.

Fred Fish Disk 85

Csh — V2.06 of Dillon's 'csh'-like shell
FileReq — Source to wildcard file requester
Hide — Hides expansion memory from programs
ImageTools — Shareware tools to manipulation IFF images
LowMem — ServerShared library to aid in low memory situations
Plot5 — A star plotting program with source.
RawID — Example of setting raw mode on standard input
Rocket — Lunary Lander for Workbench, with source.
VMore — 'more'-like text viewing utility, v1.0 with source.
Vnews — Simple Unix news reader.

Fred Fish Disk 86

AutoPointAuto-selects window under the mouse pointer, with screensaver.
ClickToFront — Double-clicks in window brings it to front, v1.1, S-E-D
Cmd — V3.0 of a tool to redirect printer output to a file.
FileIISG-Demo — Demo of Softwood File IIsg, a database manager with sound and graphics.

Fred Fish Disk 87

AdvSys — Adventure system from Byte May 1987, v1.2 E-D
AutoIconOpen — Fools Workbench to open disk icons, V1.2 update to disk 73, S-E-D
Ciaz — Converts IFF files to PostScript, V2.0, S-E-D
Commodi — tesMackrazz's Commodities Exchange, an exec library to manage the input handler, v0.4
Diff — Update to disk 75 of Unix-like 'diff', S-E-D
Dme — V1.27 of Dillon's text editor, update to disk 74 , E-D
DropShadow — V2.0 of program that puts shadows on Workbench, S-E-D
Elib — Shared library example in Manx C.
ID-Handler — An AmigaDOS device handler that generates unique identifiers, V1.0, S-E-D
Install — Alternate AmigaDOS 'install' programs, S-E-D
MemWatch — Waits for low memory trashing, V2.0, S-E-D
MovePointer — Moves pointer to given location, S-E-D
MoveWindow — Move window to given location, S-E-D
MunchingSq — Munching Squares hack, S-E-D
PalTest — Example shows test to see if this is a PAL machine, S-E-D
Sc — Generates random scenery, S-E-D
Tek4695 — Tek4695 printer driver
WBDualPF — Example of dual-playfield screen, update to disk 41, S-E-D
WarpText — Fast text rendering routines, S-E-D
Yaff — rExample IFF reader, S-E-D
Zoo — A file archiver like "arc", v1.42A, E-D

Fred Fish Disk 88 (see Fred Fish 89)

Fred Fish Disk 88 has been removed due to copyright problems

Fred Fish Disk 89 (replaces Fred Fish 80)

DirMaster — Disk catalogue program, V1.0a, E-D
FuncKey — Shareware function key editor, V1.01, E-D
MFF-Demo — Demo of MicroFiche Filer database program
ScreenShift — Adjust screen position like Preferences, S-E-D
Snake — Bouncing squiggly lines demo, S-E-D
AutoEnguirer — screen contraption requester improvement S-E-D
DemoLition — Display Hack S-E-D

Fred Fish Disk 90 (replaces Fred Fish 80)

AmiGazer — Night sky viewer of 1573 stars; set date, time, day. E-D
CardFile — AmigaBasic card file study aid. E-D
Conman — Console handler replacement gives line editing and history to most progs, v0.98,E-D
MandelVroom — Slight update to disk 78 Mandelbrot program, E-D
NewDemos — Replacements for lines and boxes demos that take less CPU time, E-D
Othello — Game of Othello, E-D
PrinText — Displays text files with gadgets, speech, IFF display, v1.2, E-D
PrtDrvGen — Automatic printer driver generator, v2.2b, E-D
RainBench — Cycles colors of Workbench backdrop or text. E-D
ShortCut — Makes single-key shortcuts for entering commonly typed CLI commands, as well as custom macros. E-D
ShowPrint — Displays and prints IFF pictures of all sizes, and controls printer output styles, v2.0 E-D
Sizzlers — Graphics demos, v1.7.0, E-D
Timer — Small Workbench timer counts time and $/ minute, E-D
Tools — Inovatronics tools: a memory editor, memory disassembler, ASCII chart, and calculator. E

Fred Fish Disk 91

Adventure Definition Language (ADL) a superset of an older language called DDL by Michael Urban, Chris Kostanick, Michael Stein, Bruce Adler, and Warren Usui. ADL enhancements by Ross Cunniff. Included are sources to the ADL compiler, interpreter, and debugger. Binaries combined by Ross with Lattice 3.03. CLI enviornment only. Documentation is available from the authors.

Fred Fish Disk 92

As6502 — portable 6502 assembler, including C source. by J. Van Ornum; Amiga port by Joel Swank
Bawk — Text processor update from FF65 Inspired by UNIX awk. Bawk searches files for patterns, performs actions based on patterns. By Bob Brodt; Amiga port by Johan Widen
HunkPad — update of FF84 version. by J. Hamilton, pads an object file to a multiple of 128 bytes for better xmodem transfer. S E
Less — Like Unix "more", better, version 1.2 update of FF74. Scrolls Back and f orward. S E by Mark Nudelman, Amiga port by Bob Levian.
Ndir — Library that implements the 4BSD unix dir access routines by MikeMeyer. S
Parse — Recursive descent expression parcer, computes, and prints expressions. includes transcendental function support. c Source included. by John Olsen
Shar — Two programs to pack and unpack shell archives (traditonal Usenet bundling of multiple text files for posting or electronic mail). Includes C source. by Fabbian G. Dufoe
SmallLib — 8 times smaller Amiga.lib replacement, binary only. by Bryce Nesbitt
UUencode — Encode/decode binary files for e mail or text-only methods. Update of FF53, includes checksum technique, compatible with older versions, plus transparent to older versions options. By Mark Horton, modified by Alan Rosenthal and Bryce Nesbitt.

Fred Fish Disk 93

Dme — Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scrolling, tite-line statistics, multiple windows, ability to iconify windows. Update of FF87, i ncludes source code. by Matt Dillon
MicroEmacs — Version 3.8i, update to FF61 includes source. Orig by Dave Conroy multiple modifications by Daniel Lawrence

Fred Fish Disk 94

AudioTools — Demo programs from Rob Peck's July/ August issue of AmigaWorld on accessing the audio device. Version 2, update of FF84. Includes source. by Rob Peck
ClickUpFront — Similar in function to ClickToFront program (FF86), bring windows to the front by clicking on any part of them. V 1.0. by Davide Cervone SE
HeliosMouse — Automatically activate a window simply by moving the mouse pointer into the window. V 1.0. Includes source. By Davide Cervone
IFF2Ps — Convert any IFF file to postscript for printing or viewing on a postscript compatible device. Version 1.2, by William Mason and Sam Paolucci E
ModulaTools — Various Modula 2 programming routines. by Jerry Mack
Terrain3d — Pseudo-random 3d relief scenery generator, update of "sc", FF87. by Chris Gray, 3d by Howard Hull

Fred Fish Disk 95

Cmd — redirects the serial.device or parallel. device output to a file. Usefull for capturing print jobs for debugging or "offline" printing. V4 By C Scheppner SE
CygnusEdDemo — Demo of CygnusSoft's CygnusEd editor, a multipeflie, multiple feature editor. also includes demo 3.0 of MandFXP. by CygnusSoft Software E
Gomf — "Get Outa My Face" makes the Guru go away and allows the user to clean-up and shutdown more cleanly. V 1.0, by Christan Johnsen E
Journal — records sequence of mouse and keyboard events, stores them in a file for future playback. Good for demos or documenting bugs. E. by D. Cervone
MergeMem — attempts merging of MemList entries of sequentially configured ram boards. When successful, allows allocating a section of memory which spans both boards. V 2, update of FF56. by Carolyn Scheppner SE
PrinterStealer — Asmilar to "Cmd", allows diversion of output destined for printer to a file. Binary only, source available from authors. by Alex Livshits and J-M Forgeas
Record-Replay — similar to "Journal", records and plays back mouse and keyboard events. binary only. source available from authors, Alex Livshits and J-M Forgeas

Fred Fish Disk 96

AnimPlayer — Animation reader and displayer by the combined efforts of Videoscape, Sculpt3D, Silver, Forms-In-Flight, and Animator.Appreciebe by Martin Hashetal.
Chess — Usenet posted Amiga port, non-Amiga interface. High playability. V 1.0, S. by J. Stanback, Amiga port by B. Leivian
Hackbench — provides source for WorkBench-Like program, for experimentation and validation of new interface ideas. Not a WBench replacement. by Bill Kinnersley
Label — Print labels with arbitray text. V 1.3, SE available from author, M.Hansen
LineDrawer — Produces line drawings based on drawing commands stored in a text file. Includes demo that draws an outline map of the USA and state borders. V1.0, SE. by John Olsen
PopUpMenu — Example code implementing pop-up menus, reasonably compatible with Intuition menus.SE. by Derek Zahn
Tek4695 — Tektronix 4695/4696 printer driver. SE. by Philip Staub
TimeRam — Fast and Chip ram test program. E. by Bruce Takahashi
WarpText — Fast text rendering routines, to be linked with application programs.Text display "as fast or faster than 'blitz". Version 2.0 update of FF87, S by Bill Kelly

INDEX OF ADVERTISERS

Fred Fish Disk 97

Replaces FF57 for Copywrite problems

CutAndPaste	Implementaions of Unix cut and paste commands. by John Weald
GraphIt	Program to plot simple functions in 2 or 3 dimensions. by Flynn Fishman
Juggler	V1.2 of robot juggler animation. Uses HAM mode and ray tracing. by Eric Graham
MouseReader	Shareware program to read text files and view IFF files using only the mouse. by William Betz
Splines	Program to demonstrate curve fitting and rendering techniques. by Helene (Lee) Taran
Shm	simple graphics demo, approximately simulates the motion of two interacting pendulums. Includes Source by Chris Ediss

Fred Fish Disk 98

Access	16 color terminal program based on Comm V1.,34. Includes Macro window, custom gadgets, colorized menus, etc. V. Beta 0.18 by Keith Young ,comm by D.J.James. E.
Backup	Writes AmigaDos disks as the backup destination. recover files from the backup disk. Requires manual decisions on disk structure. by Alan Kent SE
DCDemo	DishCat 2.3, a disk catalog program, demo limited to cataloging 100 files at a time. by Ed Alford, Micro Ace Software
HdDriver	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED
QBase	Quick-Base, a "MailBase Management utility", define and maintain a maximum of 200 records per file. by Kevin Harrise E
Thai	Thai language quiz program. Speak or type english/Thai sentences from supplied file. by Alan Kent SE

Fred Fish Disk 99

A-Render Version .3 a Ray- Tracing Construction Set for the Amiga Computer by Brian Reed ED

Fred Fish Disk 100

Berserk	Must see animation, from Sept 87 FAUG meeting. by Leo Schwab
Conman	Console handler replacement, provides line editing and command line histories transparent to any application program that uses CON: windows. Shareware V1.0 by William Hawes. E.
WBLander	Workbench display hack game, upgrade of "Rocket" on FF85, now with sound effects. By Peter da Silva. E

Fred Fish Disk 101

CrPlane	Circular plane generator for VideoScape3D. Generates a clockwise circular polygon with the specified number of vertibles. V1.0 by Thad Floryan SE
IconAssembler	Change Workbench Icons with IFF-brush files by Stefan Lindahl E
Microspell	Standalone spelling checker scans text files and reports errors. 1000 common word list, 43,000 word main dictionary wth multiple user dictionary support. Interfaces with MicroEMACS 3.9 with an emacs macro to step through the source file, stopping at suspect words and allowing the user to option. V1.0 by Daniel Lawrence, SED
Midi	midi library and utility set. Includes Midi monitor, routing utility, status utility, and more. by Bill Barton SED
PsIntrp	Postscript Interpreter reads and previews postscript files on the screen. by Greg Lee S(assy)E
StartUps	Three C startup file replacements for standard Astartup.obj and LStarup.obj. Options include (1) BothStartup.obj, for the WorkBench programs or CLI programs with or without command line parameters. (2) WBStartup.obj, for WorkBench programs or CLI programs that require no command line parameters. (3) CLIStartup.obj for CLI programs that require command line parameters but do not need to be WorkBench runnable. by Bryce Nesbitt SE

Fred Fish Disk 102

Dbug	Machine independent macro based C debugging package. Update of FF41. by Fred Fish profiling support by Binayak Banerjee SE
Match-stuff	Heavy duty text pattern matching stuff. includes simple match text replacement capability. By Pete Goodeve
Sectorama	Recover lost or damaged data from floppy or hard disks or repair a damaged volume. by David Joiner E
SilCon	Smart input line interpreter with window for full editing. Upgrade of FF50 by Pete Goodeve. E
Xicon	Use icons to call up scripts containing CLI commands. V2.0 upgrade of FF31. by Pete Goodeve E

Fred Fish Disk 103

AvlTrees	Library and test prog. implement routines for creating and using trees held in memory.S.
Calc	A programmable RPN calculator.
Cref	A C cross ref. prog. S.
DosKwik	A pair of progs. which allows you to save files to one or more floppies for quick loading. Doesn't store Dos format.
IntuiDos	A prog. to improve control and handling of the material on all disks in 'CLI-area'.
MFF-Update	A text import util. for MicroFiche Filer(demo on FF 89) and updates to some PD disk library databases.
Pack-It	Takes all files the files and dirs. on a disk and packs them into a single file. for sending via modem.
Sol	Amiga version of solitaire.

Fred Fish Disk 104

Analyticalc	Is a large and powerful spreadsheet prog.

Fred Fish Disk 105

AsmProgs	Some misc. assembly tools. In cludes some S.
BasicProgs	LeastSquare solves least square probs and graphs results. S.
Bison	A replacement for unix "yacc" command. S.
Dmouse	Another prog in the tradition of display hacks". S.
FlamKey	Allows keyboard and mouse inputs to be locked until a password is entered.
GravityWars	Game of planets,ships and black holes, v2.0 update to disk 84.
IPo2C	A util. to write a C-lang definition to mimic the intuition pointer.S
Pere-et-Fil	Ex. of creating and using reentrant processes. S.
Record Replay	Similiar to "Journal" v2.0 update to disk 95.

Fred Fish Disk 106

Funckey	Shareware function key editor, v1.1 update to disk 89. Source avail. f rom author(Anson Mah).
MoreArt	A smal selection of some Amiga artwork.
QuickFlix	An IFF slideshow and cel animation program,v0.13.
RistiNolla	A Finnish game. Also called Go-Moku. v1.0

Fred Fish Disk 107

Csh	V2.07 of Matt Dillon's csh like shell.S.
Diff	A util.,smiliar to other common "diff" programs.S.
ProSuite	Suite provides ex. code of facilites such as FileIO Requester, XText, DoRequest. and a tutorial on how to program the Amiga. Book 1.01.S
SVTools	Some useful tools. S.

Fred Fish Disk 108

AList	Dir listing prog. based on LD4 prg S
DirMaster	Disk cataloger. v1.0b, update to disk 89. S.
Dots-Perfect	Printer Driver for an Epson MX80 printer with upgrade kit installed. S.
MonIDCMP	Lets you monitor the IntuiMessages that pass through an IDCMP win dow. Prints the message class,mouse coordinates,qualifier values. Great for debugging. S.
PrintPop	A util. to send common control settings to the PRT: device. S.
Sectorama	Utilities to recover lost or damaged data from floppies and hard disks. v1.1, an update to disk 102.
Tek	Vt100 emulator for a Tektronix 4010/4014. (V2.6) update to disk 52. S.
Zoo	A file archiver, much like "arc". v1.24B. update to disk 87

Fred Fish Disk 109

Machine	A new animation.
SimCPM	A CP/M sim,simulates 8080 along with h19 emulation. S.
UUpc	Allows you to hook up your Amiga as a usenet node. S.

Fred Fish Disk 110

A68k	A 68000 assembler written in C. S.
Pdc	An optimizing C compiler for the 68000 processor. update to disk 53. but not based on the code of that disk.

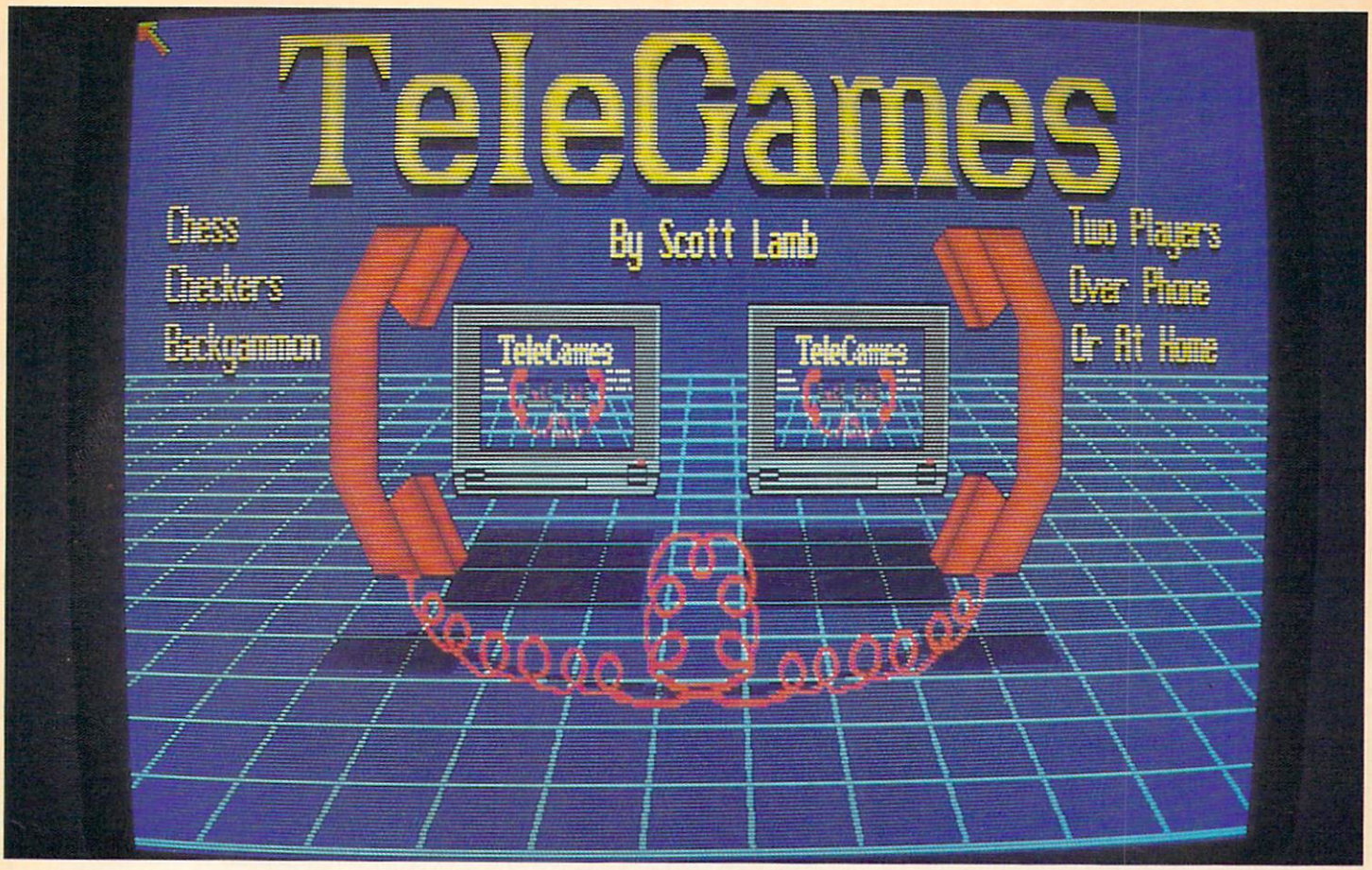
To Be Continued.....

In Conclusion

To the best of our knowledge, the materials in this library are freely distributable. This means they were either publicly posted and placed in the Public Domain by their Author, or they have restrictions published in their files to which we have adhered. If you become aware of any violation of the author's wishes, please contact us by mail.

•AC•

For PDS orders, Please use form on page 128



TeleGames is what you've waited for. The Future is here.

TeleGames allows you to use your computer and modem to play Chess, Checkers and Backgammon with a human opponent over the telephone. Only $34.95!

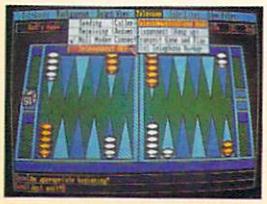
TeleGames Features

* Chess * Checkers * Backgammon
* Superb Graphic Game Simulations
* Smooth Depth Arranged Movement
* 4 angle 3D & 2D view perspectives
* Digitized Sound Effects
* Compatible with any modem
* 300, 1200, 2400, 9600 Baud
* Call originate or answer
* Null Modem Connect option
* Save Game & Transmit Game options
* Opponent File Directories
* Send and Receive Typed Messages
* Easy to Use Menus & Requesters
* All Official Game Rules Supported
* Play Over the Phone or at Home
* Legal Moves Graphically enacted on the TeleConnected computer
* Fully copyable to hard disks
* Upgrades available on our BBS

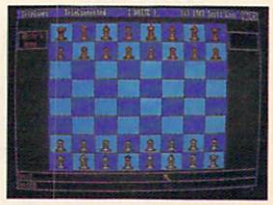
3D Chess

3D Checkers

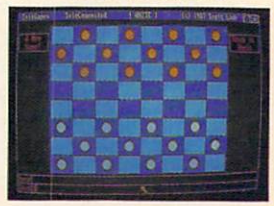
3D Backgammon



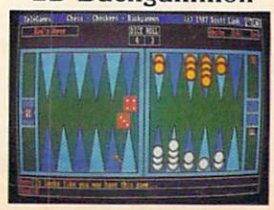
2D Chess



2D Checkers



2D Backgammon



If you Enjoy Telecomputing, You'll Love TeleGames!

Published by Software Terminal
3014 Alta Mere, Fort Worth, TX 76116
817-244-4150 Modem: 817-244-4151
Dealer Inquiries Invited

Amaze Me

Please use this order form when subscribing to Amazing Computing™, ordering Back issues, or ordering Amiga™ Public Domain Software

Name_____
Street_____
City _____ St.____ Zip _____
Amount Enclosed _____

Please circle the appropriate item: **New Subscription** **Renewal**

Please start my subscription to Amazing Computing™ with the next available issue or renew my current subscription. I have enclosed $24.00 for 12 issues in the U.S. ($30.00 Canada and Mexico, $35.00 overseas). All funds must be in U.S. Currency on a U.S. Bank

Back Issues: $4.00 each
(foreign orders add $1.00 each for Postage and Handling)

Please circle your Back issue choices below:

Vol1.1	Vol1.2	Vol1.3	Vol1.4	Vol1.5	Vol1.6	Vol1.7	Vol1.8	Vol1.9	Vol2.1	Vol2.2
Vol2.3	Vol2.4	Vol2.5	Vol2.6	Vol2.7	Vol2.8	Vol2.9	Vol2.10	Vol2.11		

Public Domain Software:
$6.00 each for subscribers (yes, even the new ones!)
$7.00 each for non subscribers

Please circle your Public Domain Software choices below:

Amicus:

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11
A11	A12	A13	A14	A15	A16	A17	A18	A19	A20	A21
A22	A23									

Fred Fish:

FF1	FF2	FF3	FF4	FF5	FF6	FF7	FF8	FF9	FF10	FF11
FF12	FF13	FF14	FF15	FF16	FF17	FF18	FF19	FF20	FF21	FF22
FF23	FF24	FF25	FF26	FF27	FF28	FF29	FF30	FF31	FF32	FF33
FF34	FF35	FF36	FF37	FF38	FF39	FF40	FF41	FF42	FF43	FF44
FF45	FF46	FF47	FF48	FF49	FF50	FF51	FF52	FF53	FF54	FF55
FF56	FF[illegible]NA	FF58	FF59	FF60	FF61	FF62	FF63	FF64	FF65	FF66
FF67	FF68	FF69	FF70	FF71	FF72	FF73	FF74	FF75	FF76	FF77
FF78	FF79	FF[illegible]NA	FF81	FF82	FF83	FF84	FF85	FF86	FF87	FF[illegible]NA
FF89	FF90	FF91	FF92	FF93	FF94	FF95	FF96	FF97	FF98	FF99
FF100	FF101	FF102	FF103	FF104	FF105	FF106	FF107	FF108	FF109	FF110

(NA denotes disks removed from the collection)

Please complete this form and mail with check or money order to:

PiM Publications, Inc.
P.O.Box 869
Fall River, MA 02722

Please allow 4 to 6 weeks for delivery

HAPPY HOLIDAYS FROM MICROILLUSIONS

THIS SEASON MICROILLUSIONS' OFFERS:

For Amiga, C64/128 and MS DOS:
FAERY TALE ADVENTURE Today's hottest game! / BLACK JACK ACADEMY / ROMANTIC ENCOUNTERS AT THE DOME.
For Amiga and soon for C64/128 and MS DOS:
LAND OF LEGENDS / PLANETARIUM / EBONSTAR / FIREPOWER / GALACTIC INVASION / TURBO.
For Amiga:
DISCOVERY and DISCOVERY EXPANSION DISKS / PHOTON VIDEO / DYNAMIC CAD / MUSIC X / DYNAMIC WORD.

microillusions™

17408 Chatsworth St., Granada Hills, CA 91344 • Inside CA 818/360-3715 • Outside CA 800/522-2041 • FAX 818/360-1464

ROME WASN'T BUILT IN A DAY, UNTIL NOW . . .

Create your own universe with SCULPT 3-D™

SCULPT 3-D brings the power of 3 dimensional solid modeling and ray tracing to the Amiga. Imagine an image: choose a color, a shape, a texture. Spin it, rotate it, extrude it into the third dimension. Pick a camera lens, set your lights, and let SCULPT 3-D create a three dimensional picture complete with shadows, reflections, and smooth shading. All in 4096 colors with true edge to edge overscan video. Easily! Automatically! Change your mind? Change the colors, textures, camera or lights in seconds and create a new image. The only limits are the boundaries of your imagination.

"I haven't had this much fun with a program since Deluxe Paint II." John Foust of Amazing Computing.

"Performance previously only available on mini and mainframe computers." Info Magazine.

Now animate your universe with ANIMATE 3-D™

Enter the fourth dimension, time. Choreograph the free flowing and simultaneous movement of objects, lights and camera through space and time. Details of object rotation, camera movements, timing and action are controlled in an easy to use graphical interface or through a simple script language. Individual objects can be linked to orchestrate complex hierarchial movements that simulate live action. Quick check wireframe playback preview's your final production: storable as a compressed animation file playable from RAM, or recorded on videotape. Additional output options include single frame VCR control or image rendering to a frame buffer card. Animations can incorporate either solid modeling or ray tracing. ANIMATE 3-D is quite simply the most powerful and easy to use animation program available for the Amiga.

Expand your universe with the BYTE BOX™

Your Amiga 500 deserves the best you can give it. More memory for more powerful applications, faster performance, better graphics, and RAM disk storage. It deserves a memory expansion system that lets you add additional memory as you need it. In easy to install and easy to afford increments. The included memory verify software provides a visual check whenever you add additional RAM. The BYTE BOX is available in a variety of configurations from 0MBytes to 2MBytes of RAM.

- Easy to install
- Fully Auto-Configure
- Fast memory that's truly fast
- Has its own power supply
- Fully tested and ready to use
- Zero wait state design
- Low profile case
- Memory check software

BYTE by BYTE™
CORPORATION

Aboretum Plaza II 9442 Capital of Texas Highway North Suite 150 Austin, TX 78759 (512) 343-4357